Titre: Title:	Transport protocol for a real-time communication in wireless sensor actor networks - WSAN
Auteur: Author:	Paul Stoina
Date:	2008
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Stoina, P. (2008). Transport protocol for a real-time communication in wireless sensor actor networks - WSAN [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8274/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/8274/
Directeurs de recherche: Advisors:	
Programme: Program:	Non spécifié

UNIVERSITÉ DE MONTRÉAL

TRANSPORT PROTOCOL FOR A REAL-TIME COMMUNICATION IN WIRELESS SENSOR ACTOR NETWORKS – WSAN

PAUL STOINA
DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INFORMATIQUE)
AVRIL 2008

Your file Votre référence
ISBN: 978-0-494-41579-5
Our file Notre référence
ISBN: 978-0-494-41579-5

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

TRANSPORT PROTOCOL FOR A REAL-TIME COMMUNICATION IN WIRELESS SENSOR ACTOR NETWORKS – WSAN

présenté par: STOINA Paul

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

Mme. CHERIET Farida, Ph. D., présidente

M. QUINTERO Alejandro, Doct., membre et directeur de recherche

M. SAMUEL Pierre, Ph. D., membre

I dedicate this work to Irina, Thea and Mihaela

Acknowledgements

First of all, I wish to express my sincere gratitude to my supervisor Mr. Alejandro Quintero, Professor in Department of Computer Engineering at École Polytechnique, Université Montreal, who gave me guidance, support, perceptive advices, and encouragement into my research and completion of this work.

I would like to acknowledge and extend my heartfelt gratitude to Mr. Ernesto Hidalgo, M.Sc.A, who gave me, with enthusiasm and inspiration, different suggestions along my research.

I wish to express my thanks to my work colleagues: Mr. Eric Robin, Mr. Daniel Tremblay and especially for Mr. Pierre Gosselin, my superior. They have offered me the appropriate support in order to have a flexible program and follow my studies.

I wish to address special gratitude to Abbé André Blais, who encouraged and sustained me morally since I started my study.

My research was made also possible due to Mr. Ray James Bernard, Professor at Concordia University Montreal who motivated me during the most crucial moments.

I want to express my gratitude to Mrs. Mariana Jemna who reviewed the thesis text and gave me suggestions that clarified it from the language point of view.

Last but not least, I owe a great deal to my Romanian friends who encouraged my efforts since the beginning: Mrs. Madalina Suceveanu, Access Network Manager at Orange-Romania Company and Mr. Sorin Barbulea, Professor at University Polytechnica Bucuresti, without whom it wouldn't have been possible to start this important project.

Thus, I greatly appreciate the time that all the above-mentioned spent to understand my motivation during the last four years.

Abstract

The wireless sensor and actor applications need data that arrive in time and which are necessary for actors to act during a validity time. This demand is fulfilled by a real-time communication for applications that utilize WSAN technology.

The protocols TCP and UDP were not designed for a real-time communication and for a wireless network. The designed protocol – Bonaventura – can manage the data units delay taking into account time and space evaluation. This transport protocol has an entity in every node along the data units' path.

The scheduling mechanism of the protocol determines a data unit priority, which establishes the transmission order of the data units in a node. The traffic shaping mechanism limits the data unit arrival rate to keep the network without congestion. These two mechanisms help the data units to meet their deadline.

The protocol designed in this work was compared with UDP protocol using the mathematical models in MATLAB. The two protocols were compared for the data units travel times and for the timelyput: the throughput of the data units that arrive in time and could be used by the application. It is a protocol with better results – more than 18% – for a 500 sensors and 3 sinks network deployed on a surface of 1000 m by 1000 m and using different values of the transmission rate.

Résumé

Les applications pour les réseaux de capteurs et robots sans fil ont besoin des données qui arrivent à temps, données nécessaires pour les robots afin d'actionner dans un temps dans lequel les données restent valides. Cette demande est réalisée par une communication en temps réel pour les applications qui utilisent la technologie WSAN.

Les protocoles TCP et UDP ne sont pas conçus ni pour une communication en temps réel, ni pour un réseau sans fil. Le protocole conçu, Bonaventura, peut gérer le délai des données en tenant compte d'une évaluation spatio-temporelle. Ce protocole de transport a une entité dans chaque nœud sur la route suivie par les données entre le capteur générateur et la passerelle (« sink »).

Le mécanisme d'ordonnancement du protocole détermine la priorité de données et celle-ci établit l'ordre de transmission des données d'un nœud à l'autre. Le mécanisme de canalisation du trafic limite le taux d'arrivées de donnés pour préserver le réseau sans congestion. Les deux mécanismes aident les donnés d'arriver avant une limite imposée par l'application – l'échéance.

Le protocole conçu a était comparé au protocole UDP en utilisant des modèles mathématiques dans MATLAB. Les deux protocoles ont été comparés pour le temps parcours (temps de traverser le réseau) et pour le débit des données qui arrivent avant l'échéance, nommé « timelyput ». Le protocole Bonaventura a des résultats supérieurs - de jusqu'à 18% comparativement à l'UDP pour un réseau de 500 capteurs et trois passerelles, déployé sur une surface de 1000 m sur 1000 m en utilisant différentes valeurs de débit binaire.

Condensé en français

Introduction

Un capteur est un microsystème embarqué capable de mesurer un phénomène physique dans l'environnement et de transmettre le résultat de la mesure à l'aide d'un système de communication sans fil. Plusieurs capteurs peuvent être déployés dans une zone spéciale dans laquelle on doit superviser un phénomène physique. Ces capteurs forment alors un réseau. Les messages avec l'information sur le phénomène observé vont passer d'un capteur à l'autre afin d'être rapportés à un « sink » ou à un « acteur ». Le « sink » est une passerelle ou les données générées par le réseau de capteurs doivent être rapportées pour être acheminées vers les « acteurs » ou vers le centre de surveillance. Dans ce contexte, les « acteurs » sont des robots mobiles qui reçoivent l'information du réseau de capteurs et qui agissent afin de réaliser une tâche conformément à l'application. Les robots forment eux aussi un réseau mobile qui communique sans fil. Entre les dispositifs décrits il faut avoir une communication sans fil et, par conséquent, une pile de protocoles afin de réaliser cette communication.

Le phénomène physique surveillé par le réseau de capteurs peut se modifier et alors les données envoyées par les capteurs ont un intervalle de validité. Une information qui n'est plus valable est inutilisable. En plus, si l'information est utilisée par les robots, ils doivent réaliser la tâche avant que les caractéristiques du phénomène sur lequel ils agissent, changent. Alors, il se forme une chaîne d'actions: le phénomène physique est mesuré par le capteur, l'information est préparée par le capteur, l'information est transmise tout au long du réseau de capteurs, l'information arrive directement ou via passerelle vers les robots, les robots décident ensemble de la meilleure action à suivre, les robots réalisent l'action sélectionnée. Comme toutes les actions qui composent ces chaînes doivent être limitées en temps, la communication capteurs-passerelle ou capteurs-robots doit être également limitée en temps. Ce type de communication est une communication en temps réel.

L'objectif de ce travail est la conception d'un protocole de transport pour une communication en temps réel dans un réseau de capteurs.

La communication en temps réel est une communication dans laquelle le temps de transport des données de la source à la destination est critique. Pour ces données existe un temps limite de communication dénommé échéance. Ce type de communication nécessite un protocole de transport spécifique.

Le temps de transport est le temps durant lequel une donnée est prise en charge par le protocole dans le nœud source et elle est remise au niveau application par le protocole de transport à la destination. Nous allons nommer ce temps durée du parcours ou temps de parcours. Tout au long de la route les données passeront d'un nœud au nœud suivant le réseau. On va nommer ce temps: délai de saut. Le nœud source peut être n'importe quel capteur dans le réseau. Le nœud destination peut être la passerelle ou un robot. Le temps de parcours est plus court quand la donnée est remise à un robot, mais, comme les robots sont mobiles, il est possible qu'ils ne se trouvent pas aux alentours des capteurs. Dans ce cas, la communication entre capteurs et robots utilise la passerelle. Donc, nous essayerons de limiter le temps de parcours entre les capteurs et la passerelle.

Nous avons choisi deux métriques de performance pour le protocole de transport : le temps de parcours et le débit de données qui arrivent à temps (avant l'échéance) à la couche application du nœud destination. Ce débit a été nommé « timelyput ».

Protocoles de temps réel dans les documents de présentation technique

Pour la deuxième couche nous avons évalué deux protocoles : « An Implicit Prioritized Access Protocol for Wireless Sensor Network » et « A Robust Implicit Access Protocol for Real-Time Wireless Collaboration ». Les deux protocoles utilisent le mécanisme « Earliest Deadline First – EDF » qui prend en considération l'échéance des données. Le premier protocole a comme inconvénients :

- réseau de capteurs qui doit être synchronisé;
- système de communication sans fil complexe pour un capteur (plus qu'un canal de transmission);

- une architecture de réseaux complexe;
- un seul point de défaillance qui est le capteur élu comme routeur.

Le deuxième protocole a deux améliorations en le comparant au premier protocole: le mécanisme de rétablissement et le budget de la bande passante. Le principal désavantage est qu'il nécessite une connectivité complète des nœuds qui implique une grande couverture pour le système de communication de capteurs (réseau non-scalable).

Pour la troisième couche nous avons analysé trois protocoles de routage « Directed Diffusion : A Scalable and Robust Communication Paradigm for Sensor Networks », « Greedy Perimeter Stateless Routing for Wireless Networks (GPSR) » et « A Spatiotemporal Communication Protocol Network for Wireless Sensor Networks - Speed ». Le premier protocole a été pris en considération parce qu'il tient compte de la nature du réseau de capteurs et parce qu'il peut être adapté pour une communication en temps réel. Mais le protocole prévoit l'agrégation des données au long de la route. Une évaluation sur le délai de l'agrégation des données a été faite dans l'article [6]. Celui-ci montre que cette agrégation a un impact négatif sur le temps de parcours. Le deuxième protocole a été analysé parce qu'il utilise la location géographique (qui est dans la nature des capteurs) pour routage mais il ne tient pas compte du délai des données pour une communication en temps réel. Le troisième protocole – Speed – assure un délai sur une distance donnée. Si, par application, on impose une vitesse pour le déplacement des données, alors, le délai est directement proportionnel à la distance parcourue. Il est également un protocole robuste mais, comme inconvénient, il est complexe et cela implique des ressources qui sont limitées.

Pour la couche transport nous avons évalué les protocoles TCP et UDP. Le protocole TCP est un protocole de transport excellent conçu pour le réseau avec fil, mais il ne tient pas compte du temps du transport des données. Ses performances diminuent quand il est utilisé dans une technologie sans fil. En conclusion, il ne convient pas à un réseau de capteurs. En plus, TCP est un protocole très complexe qui ne convient pas aux capteurs qui sont pauvres en ressources.

UDP est un protocole de transport simple qui n'utilise pas les mécanismes de TCP. En raison de cela UDP est meilleur que TCP quand il s'agit d'un réseau de capteurs. Mais, comme le TCP, le UDP ne garantit pas le temps de transport limite pour les données.

Un protocole de transport est un protocole qui a des entités seulement en deux nœuds : source et destination. Il est intéressé uniquement aux paramètres de transport d'un bout à l'autre de la communication. Nous pouvons dire qu'un protocole de transport est un transport à une orientation spatiale. Mais, un protocole en temps réel doit utiliser les couches qui se trouvent en dessous pour remettre les données dans un temps limité. Par conséquent, il faut changer l'orientation spatiale en une orientation temporelle du protocole.

Comme une unité de données est un élément passif (elle ne peut pas s'accélérer d'elle-même) il faut utiliser des mécanismes temporels. Si nous adaptons un protocole de transport avec des mécanismes temporels comme l'ordonnancement des données basé sur priorité et la filtrage des données, à cause d'une orientation spatiale du protocole de transport, ces mécanismes ne peuvent pas changer beaucoup dans l'amélioration du temps de parcours des données. Il faut changer l'utilisation de ces mécanismes globalement, d'un bout à l'autre, dans une utilisation locale, dans chaque nœud, tout au long de la route.

Protocole - conception

L'idée fondamentale de ce travail est d'implémenter un protocole de transport ayant une entité dans chaque nœud tout au long de la route des données. Cette entité – entité temps - aura des mécanismes de temps réel. Dans ce cas l'évaluation pour prendre des décisions au niveau transport va être faite localement et non globalement comme est maintenant le cas du protocole de transport. Cela changera l'orientation spatiale du protocole en une orientation temporelle. Le lieu de l'entité de temps du protocole de transport se trouve dans la quatrième couche à cause de:

- l'utilisation du mécanisme d'ordonnancement doit être faite sous la couche application: les données qui sont générées par le capteur doivent, aussi, entrer dans ce mécanisme
- les données doivent être premièrement ordonnées pour être envoyées vers les couches inférieures et puis elles doivent être acheminées vers le nœud suivant.

Les deux affirmations concluent que l'entité de temps de protocole doit être dans la quatrième couche.

Modèle

La densité des capteurs sur l'unité de l'aire est un paramètre très important dans le déploiement du réseau. Si les capteurs sont très éloignés, ils ne peuvent communiquer entre eux et le réseau ne peut fonctionner. Si les capteurs sont très proches l'un de d'autre, alors ils vont attendre longtemps pour accéder au canal (qui est unique) pour transmettre la trame puisque le canal va être occupé presque tout le temps par les capteurs voisins. Après le déploiement des capteurs, ceux-ci vont avoir une densité de surface a_N, et nous pouvons estimer le nombre moyen m de capteurs qui se trouvent dans un voisinage d'un capteur. Tous les capteurs dans un voisinage sont influencés par la transmission d'un capteur.

Nous avons considéré que tout nœud dans un voisinage a une file de transmission, que le taux des arrivées des données est λ pour chaque capteur, et que les processus sont indépendants l'un de l'autre. Alors nous avons modélisé toutes les files de transmission par le modèle de file M/M/1. Tous les capteurs formeront une file virtuelle avec le taux des arrivées de $\lambda_M = m*\lambda$ *(m* nombre de nœuds dans un voisinage). Dans ce cas, le délai à chaque saut est donné dans l'équation Eq.3.6. Ce délai inclut le temps d'attente dans la file virtuelle et le temps de transmission vers le nœud suivant. Le taux maximal des arrivées pour un capteur est donné dans l'équation Eq.3.8 et il est obtenu pour un délai infini du saut infin.

Nous avons calculé le taux de génération des données pour chaque capteur - Eq.3.15 (égal pour tous les capteurs du réseau) – en estimant que la génération des unités des données pour tout le réseau ne peut pas être plus grande que celle que les passerelles peuvent transférer.

En considérant le taux des arrivées pour chaque capteur tout au long de la route comme constant nous avons estimé le temps de parcours maximal comme dans l'équation Eq.3.12. Mais, si on considère un taux des arrivées qui peut varier entre le taux de génération (dans les zones marginales du réseau) jusqu'à un taux λ (très proche de la passerelle), nous pouvons estimer un temps moyen du parcours comme dans l'équation Eq.3.20. Les courbes de temps maximal et moyen de parcours en fonction de la variation des taux des arrivées sont montrées dans la figure 3.12.

Mécanismes

Pour l'entité temps du protocole, nous avons choisi, pour être implémentés, trois mécanismes de temps: filtrage, ordonnancement à base de priorité et canalisation du trafic.

Le mécanisme de filtrage va rejeter toutes les données qui dépassent l'échéance.

Le mécanisme d'ordonnancement changera l'ordre de transmission des unités afin de favoriser les unités qui sont en retard et donc faciliter leurs arrivées à temps au nœud de destination. Pour changer l'ordre de transmission des données nous devons évaluer chaque unité de donnée, de lui attribuer une priorité et de l'envoyer vers la couche inférieure en fonction de la priorité reçue. L'évaluation de chaque donnée est réalisée par la fonction « priority » ζ qui prend des valeurs $1 \div 7$ correspondant à la priorité donnée. La fonction ζ analyse chaque donnée du point de vue spatial et temporel. Il ne suffit pas d'évaluer seulement temporellement (si elle est encore valable) une unité de données mais, également, elle doit être évaluée sur la distance parcourue, entre l'endroit de sa génération jusqu'à l'endroit d'évaluation. Nous avons divisé la surface d'évaluation en sept zones correspondant aux valeurs de priorité attribuées à chaque unité de données. Si une donné se trouve dans la zone 1 elle est très retardée et elle prend la priorité 1. Elle doit être envoyée immédiatement. Dans la figure 3.20 il y a une représentation tridimensionnelle de la fonction « priority » ζ. Après l'évaluation d'une donnée du point de vue spatio-temporel, elle est mise dans un système de files; le nombre des files correspond au nombre des valeurs que la fonction « priority » ζ prend. La priorité correspond au numéro de la file dans laquelle la donnée est mise. L'entité du temps va transmettre les unités des données en commençant avec la file plus prioritaire (dans ce cas la file numéro 1). L'envoi des unités suivra une politique non préemptive : si, pendant une transmission d'une unité de donnée va apparaître une autre donnée dans une file plus prioritaire, la donnée plus prioritaire va attendre jusqu'à la fin de la transmission commencée.

Le mécanisme de canalisation du trafic va assurer un taux des arrivées des données afin que le réseau n'entre pas dans une phase de congestion. L'algorithme de ce mécanisme s'appelle seau à jetons.

Comme le mécanisme de filtrage peut être intégré dans le mécanisme d'ordonnancement, les parties constituantes de l'entité du temps du protocole de transport conçu sont : le mécanisme ordonnancement – filtrage et le mécanisme de canalisation. Les deux mécanismes partagent le même système de files.

Modèle analytique

Le modèle analytique est construit pour l'évaluation des performances du protocole conçu: Bonaventure. Les métriques de performances sont : le temps de parcours des unités des données et le débit des données qui peuvent être utilisées par l'application (des donnés qui ont le temps de parcours plus court que l'échéance). Nous avons nommé ce débit « timelyput ». Le protocole conçu Bonaventura - BVP a été comparé au protocole UDP.

Nous avons analysé le temps de parcours et les principales composantes sont les délais pour chacun saut tout au long de la route et les temps de traitement à la couche transport. Les différences de temps de parcours entre les deux protocoles sont :

- les délais de saut pour UDP sont plus larges parce que le taux des arrivées des données n'est pas limité comme dans le cas du protocole BVP.
- les temps de traitement à la couche transport sont plus larges pour BVP parce que les données sont traitées dans chaque nœud au long de la route et parce que le protocole BVP est plus complexe que UDP.

Mais les délais de saut sont plus larges que les temps de traitement même si le débit binaire de transmission (*W*) est grand (ce qui implique que la transmission d'une unité de donnée est très court). En conséquence, les temps de traitement ont été négligés dans le modèle d'analyse.

Nous avons analysé le « timelyput » qui représente le débit des unités des données à temps. Si nous avions pris comme métrique de performance le débit de toutes les données arrivées à la passerelle, alors le UDP se serait comporté mieux que le BVP. Mais, pour une communication en temps réel, on va utiliser seulement les données qui arrivent à temps (avant l'échéance) le soi disant « timelyput » qui est la deuxième métrique de performance.

Pour construire le modèle mathématique, nous avons approximé le « timelyput » par le numéro des unités des données qui arrivent en temps. Nous avons approximé le débit par le nombre des unités de données générées par les capteurs en considérant que toutes les données générées arriveront aux passerelles. Par conséquent, si nous voulons évaluer le « timelyput, il faut comparer le nombre des données arrivées à temps des deux protocoles.

Pour calculer le temps de délai nous avons eu besoin des taux des arrivées dans chaque capteur, qui a été considéré comme égal à la somme du taux de génération des données et un taux de transit (interne) des données du chaque capteur. Ce taux interne – τ prend une valeur aléatoire entre 0 et le maximum du taux des arrivées calculé pour le réseau.

Les limites du modèle sont :

- le calcule du délai de saut est basé sur la supposition que toutes les nœuds ont le même taux des arrivées
- la supposition que le routage de données dans le réseau est statique
- le rejet des données non valides par les mécanismes de BVP dans les nœuds de transit n'a pas été modélisé
- le rejet des données lorsque les mémoires tampon sont pleines n'a pas été modélisé quand le protocole UDP est utilisé

- le temps de traitement dans la couche de transport est négligé
- le mécanisme de canalisation pour BVP n'a pas été modélisé
- la fonction τ qui représente le taux des arrivées internes (de transit) de données dans chaque nœud est modélisé par une fonction aléatoire

L'implémentation du model analytique a été fait avec le MATLAB 7.1. Nous avons conçu trois programmes :

- un programme pour générer un réseau de 500 capteurs et 3 passerelles dans un champ carré de 1000 m sur 1000 m. Le rayon du système de communication radio est de 100 mètres. Le déploiement des capteurs a été fait aléatoirement et le résultat est dans la figure 4.6. Le routage des données dans le réseau est statique: si un capteur a plusieurs nœuds dans son voisinage, ce capteur envoiera leurs données au capteur le plus proche de passerelle (et qui se trouve dans le voisinage).
- un programme pour évaluation de « timelyput » en deux variantes : 1. les courbes: « timelyput » en fonction de l'échéance pour un taux de génération fixe et 2. les surfaces: « timelyput » en fonction de l'échéance et du taux de génération des données.
- un programme pour évaluation du temps de parcours sous la forme de deux histogrammes.

Chaque capteur génère 10 000 unités des données. Nous avons étudié trois scénarios :

- un réseau des capteurs à un faible débit binaire : 64 kbit/s
- un réseau des capteurs avec un moyen débit binaire: 512 kbit/s
- un réseau des capteurs à un fort débit binaire: 2048. kbit/s

Pour le premier scénario (dans lequel nous avons utilisé un débit binaire faible) nous avons observé que le réseau a de larges temps de parcours de données parce que le temps de transmission d'une unité de donnée est de 16 ms. Le débit impose aussi de petites valeurs de taux de génération des données dans chaque capteur. Pour un taux de génération de 5 unités des donnés/s le réseau est en totale congestion et le protocole

UDP ne peut plus fonctionner, mais le protocole BVP peut avoir une « timelyput » de 400 000 unités qui représente 8% du débit de données générées. La différence entre les deux protocoles est infime quand les ressources du réseau ne sont pas dépassées, mais la différence augmente quand les ressources sont utilisées en pleine capacité ou dépassées.

Pour le scénario dans lequel nous avons utilisé un fort débit binaire nous avons observé que le réseau donne de très court temps de parcours de données car le temps de transmission d'une unités de donnés est de 0.5 ms. Aussi, le débit binaire permet de grandes valeurs de taux de génération des données dans chaque capteur. Pour un taux de génération de 150 unités des donnés/s, le réseau est en totale congestion et le protocole UDP ne peut plus fonctionner, mais le protocole BVP peut avoir une « timelyput » de 400,000 unités qui représente 8% du débit de données. Aussi, la différence entre deux protocoles est petite quand les ressources de réseau ne sont pas dépassées, mais la différence augmente quand les ressources sont dépassées.

Pour le réseau des capteurs avec un moyen débit, les résultats ont été présentés dans l'annexe 2. Les conclusions sont analogues aux scénarios présentés.

Pour cette évaluation avec les deux métriques de performance nous avons conclu que BVP peut avoir des performances plus grandes, jusqu'à 18 %, que UDP.

Conclusions

Cette thèse veut présenter un protocole de transport pour une communication en temps réel dans un réseau de capteurs. Le protocole conçu – Bonaventura – veut changer la vision des protocoles de transport qui ont des entités seulement dans les extrémités de la communication. Pour gérer une communication en temps réel, un protocole de transport doit avoir des entités dans chacun nœud tout au long de la route. Le protocole a été comparé à UDP et nous avons tiré les conclusions suivantes:

- les entités du temps du protocole gèrent le temps de chaque unité donnée afin qu'elle accomplisse les conditions de la communication en temps réel
- le protocole Bonaventura peut être utilisé aussi pour le réseau de capteurs et robots sans fil

- l'entité du temps est implémentée à l'aide d'un mécanisme d'ordonnancement et filtrage des données pour évaluer chaque unité de donnée en deux dimensions, temps et espace, afin de changer l'ordre de transmission des données et de favoriser les données en retard
- l'entité du temps est implémenté aussi avec un mécanisme de canalisation de trafic pour limiter le taux des arrivées des donnés dans chaque capteur et, en conséquence, pour empêcher l'entrée du réseau en congestion
- le coût pour avoir une communication en temps réel est le temps. Le protocole Bonaventura est plus complexe que UDP et les traitements de données dans l'entité de chaque nœud au long de la route donnent des délais dans la quatrième couche pour BVP plus élevés que pour UDP. Mais ce désavantage est bien compensé par les délais de saut entre les nœuds plus petits dans le cas du protocole Bonaventura.
- la comparaison des deux protocoles a démontré que le comportement du BVP et de jusqu'à 18 % meilleur que le comportement de UDP.
- des recherches futures peuvent être réalisées : estimations des taux de génération et des taux des arrivées des données dynamique; modification de la limite imposée pour éviter la congestion dans le réseau à partir de la condition du canal de transmission; autres paramètres d'évaluation des données que le temps et l'espace, en conformité avec l'application; une police de rejet des unités des données plus flexible.
- l'amélioration de l'évaluation des performances, en simulant le protocole Bonaventura dans Network Simulator 2 (NS-2) ou dans un réseau de capteurs réel.

Table of Contents

Table of Figures

List of Tables

List of Acronyms and Abbreviations

AODV	Ad-hoc On demand Distance Vector routing
BPR	Back Pressure Rerouting
BVP	BonaVentura Protocol
CPU	Central Processor Unit
CTS	Clear To Send
DCF	Distributed Coordination Function
DSR	Dynamic Source Routing
EDF	Earliest Deadline First
EFLN	Explicit Link Failure Notification
et al.	et alii (and others) – Latin abbreviation
FDM	Frequency Division Multiplexing
FIFO	First In First Out
FS	Forwarding candidate Set
GPSR	Greedy Perimeter Stateless Routing
LAN	Local Area Network
LAP	Location Addressed Protocol
LEACH	Low Energy Adaptive Clustering Hierarchy
MAC	Medium Access Control
MANET	Mobile Ad-hoc NETwork
MEMS	Micro-Electro-Mechanical-System
M/M/1	Markovian distribution for arrival process / Markovian distribution for service process / servers number – queue notation
NFL	Neighborhood Feedback Loop
NGF	Nondeterministic Geographic Forwarding
NS	Neighbor Set
PDMR	Packet Deadline Miss Ratio

RAP	Real-time communication Architecture for large scale wireless sensor network
RTC	Real-time Transport Protocol
RTO	Retransmission TimeOut
RTS	Request To Send
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
WSAN	Wireless Sensor Actor Network
WSN	Wireless Sensor Network

List of Annex

Chapter 1 Introduction

The advent of the micro-embedded system has made possible new and interesting applications. A micro-embedded system could be utilized to sense a physical phenomenon and thus it becomes an embedded sensor. Many embedded sensors communicating between them form a sensor network that could have a large sphere of applications. As consequence a great effort of research points to develop the micro-embedded systems, sensor networks and applications. The development of the sensor networks and their applications is not possible without the development of the communication system and its protocols. The special characteristics of the sensor networks forced to develop protocols different from those utilized in the wire line world or in the mobile ad-hoc networks. Moreover, as the applications of the sensor networks have a real-time characteristic, a real-time system communication has to be developed in order to support these applications. In consequence the delay metric has to be taken into account when the protocols are designed. The scientific literature shows some proposals on protocols for data link and networking, but little work has been made for a delay aware transport layer protocol.

In this chapter we present the main concepts and requirements for the sensor networks, the statement problems of real-time communication protocol and the research objectives

1.1 Definitions and the Basic Concepts

The integration of mechanical and electronic elements in Micro-Electro-Mechanical Systems (MEMS) has given the possibility to build micro-systems that can sense or control the environment.

A sensor is a micro-embedded system that makes a multifunctional monitoring of the environment: it measures mechanical, thermal, biological, chemical, optical, or magnetic phenomena. The sensor components are shown in the figure 1.1 [13]. To communicate the sensed data, the sensor has a wireless system communication. In order to meet the demand of small size and low cost for a sensor, the capabilities of sensing, data processing and communicating are made by scarce resources like power supply, memory, processing power, operation system, or system communication.

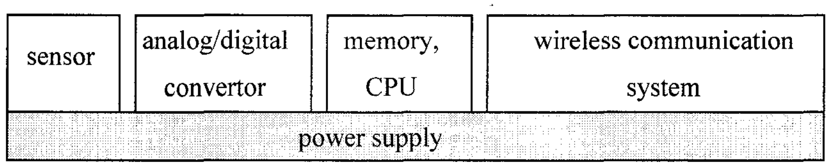


Figure 1.1 The sensor components

Many sensors, deployed over a large area that has to be sensed, form a wireless network where the nodes are the sensors which detect the physical phenomenon, carry out simple computations and communicate to report the sensed data. This sensor network, deployed over an area, supports a specific application is called wireless sensor network. – WSN. The sink or base communication is the gateway where the data are reported and from where the queries about the sensed phenomenon are spread over the network.

For the applications that not only sense the environment but also perform specific tasks, there are devices richer in resources called: actuators, actors or robots. We will adopt the name of actor. In this case the actors become gateways that spread queries or receive the gathered information under the central base unit management. The actors have to be very mobile to make quick and accurate actions.

The set of sensors and the set of actors could be regarded as two different networks or as a single network called wireless sensor and actor network –WSAN. The components of the wireless sensor actor networks are shown in the figure 1.2.

The sensors on one side and the actors on the other side must be coordinated between them in order to maintain the networks. The actors send to the sensors the queries that describe the phenomenon that has to be monitored. The sensors, based on the actors' demands, send the data concerning the environment. The actors, based on the information gathered by the sensors decide the actions that have to be taken in concordance with the application for which the system was conceived.

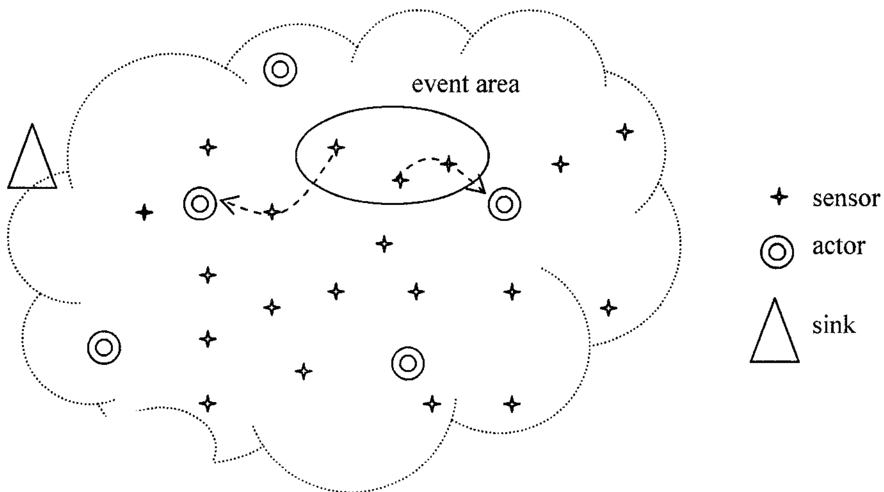


Figure 1.2 Wireless sensors actor network components

The consequence of the action of the actors or natural causes make the monitored phenomenon to change in time and therefore the information gathered by the sensors has to be valid when the actors do the task. Therefore all the chain must be a real-time system which implies that the communication should be a real-time.

All these make the communication system a vital resource for sensor and actor networks which issue a strong challenge: excellent performances with scarce resources.

1.2 Statement

The idea to use embedded systems for sensing the environmental phenomenon was very fascinating but it raises many challenges. One of all these challenges is that the information gathered by the sensors has to be transmitted in an efficient manner. The principal differences between the sensor networks and ad-hoc networks made that the protocols developed for ad-hoc networks cannot be used for sensor networks with the same performances. Therefore a large amount of research work was used to find efficient protocols that convey the information between sensors and actors. But the applications designated for sensor networks that monitor a physical phenomenon are real-time applications. Lately, the research work was focused to design protocols that support real-time application, thus a real-time communication.

The problem treated in this thesis is to design and to build a real-time transport protocol for the sensor networks. The requirements analysis, for the requirements of the real-time applications and for the real-time protocols proposed in the literature, lead us to the following statements:

1. channel condition: The functioning of the applications and the protocol stack have to be adapted to the channel condition. In the wire line world, the applications were served by the lower layers. These layers hide the network, data link and physical conditions and try to serve the application and did not take into account the delay to accomplish the task. In the sensor world, since the main importance is the transmission delay, the application and the protocol stack have to adapt their functioning to the channel condition. When the channel condition is bad, it is useless for the application to demand to the protocol stack to send information. The only effect to this is to delay the data which could become invalid. This adaptation at the channel condition will be made using cross-layer design.

2. no data aggregation: In the literature, the sensors have to make data aggregation in order to build a more accurate report or to remove redundant data (for congestion avoidance). But the data aggregation consumes energy and uses computation that is directly related to the message delay. It was shown that the aggregation introduce

a variable delay. Therefore, it will not use the data aggregation on the information gathered by the sensors. When the actors receive the gathered information they will compute and coordinate between them to take the appropriate action. The actors are richer in resources and the action will be made with a lesser delay.

3. congestion control: To achieve a low and a bounded delay it has to have a strict control of the congestion. To avoid the congestion it has to fulfill three requirements: i) to maintain the sensor at a optimum load; ii) to maintain a network without congestion; iii) to maintain the link without collisions. These requirements have to be fulfilled simultaneously.

4. local than global: The control of the sensor network has to be made from the local point of view rather from the global one. It is not necessary to wait till the message arrives to the destination to in order to take the appropriate action to correct the network behavior. This global management derives delays. The control of the network has to be made local: local verification and actions that prevent unbounded delays.

5. sensor load: The sensor load should be devised in the load to convey the other sensor messages and the load to transmit own messages. A metric is necessary in order to measure those loads and a strategy to balance between them. It is useless to try to send a message to a sensor node that is almost fully loaded in conveying the other nodes messages. The sensor cannot transmit its own messages, and the conveyed messages will be delayed.

These problem statements summarize the points of view expressed in this thesis, in order to design and implement a real-time transport protocol for sensor networks. The main difficulty in this task is that the sensors are devices with scarce resources making it a challenge research.

1.3 Research Objectives

The main objective of the research is to design a transport protocol for sensor networks to support real-time applications therefore a protocol capable to deliver the data timeliness. The protocol must deliver the information gathered by the sensors in time, assuring its validity when the application uses it. The protocol has to be independent in terms of real-time network capability in order to respond to any proposed application. The thesis will not take into account the security problems. More precisely this thesis has the following objectives:

- Analysis the requirements of the real-time applications which use sensor networks and the sensor networks capability in terms of real-time.

- Analysis the proposed sensor network protocols in the literature. It will be outlined the strong and week points for these protocols relative of their timeliness capability.

- Design a transport layer protocol for real-time communication for sensor networks. We will find the appropriate metrics and mechanisms and develop the algorithms.

- Evaluate the performances of the protocol and its mechanisms using the mathematical model.

- Compare the protocol performance with the transport layer protocol used now for sensor networks.

1.4 Work Outline

The thesis is structured as following:

In Chapter 1, are described the basic concepts of sensor, sensor networks, actor and wireless sensor and actor networks. It is also presented the concept of the real-time communication.

Chapter 2 presents a survey on the scientific literature for the real-time communication in the WSAN: a. real-time application requirements and b. real-time protocols proposed in the literature; their strong and weak points concerning the timeliness.

Chapter 3 makes the analysis of sensor networks, the design of the protocol and the implementation of its mechanisms.

Chapter 4 shows the mathematical model and implementation; evaluation and the protocol limits; the comparison of the design protocol with another transport protocol.

Chapter 5 presents the conclusions of this work and the directions for further research.

Chapter 2 Real-Time Communication in Sensor Networks

In this chapter we present, based on the literature review some aspects about the real-time communication requirements and about important time aware protocols for sensor networks designed for data-link and network layer. For transport layer we present the TCP and UDP protocols and we try to evaluate these protocols adapted with simple time aware mechanisms.

2.1 Real-Time Communication Requirements

We intend, to outline the main characteristics of the sensor network and clarify some real-time communication aspects such as: real-time communication types data delivery time or travel time and performance metrics.

2.1.1 Characteristics of Sensor Networks

The main characteristics and their implication for sensor networks are the following:

• Real-time communication. This is the most important characteristic of the sensors to actors (sinks) communication. Usually, the sensors gather information over zones where it is difficult for persons to reach. Therefore, sensors are deployed in an unattended manner. After the deployment, the sensors receive queries relating to the sensed phenomenon. Based on the queries, the sensors gather the information. Gathering information about a physical phenomenon is the common aspect of all applications: This information is reported to a central base unit via a sink. Usually, at the central base unit the application build reports about the monitored the phenomenon. The information is sent toward the sink using the communication between sensors. Thus the sensors are sources of information and nodes that transit the information up to the sink. As the physical phenomenon changes, the gathered information has a limited validity. The validity of the information depends on how rapid is the changing of the phenomenon. As the information has a limited validity, the time to arrive at the central base unit is critical and should be as short as possible.

At the central base unit, based on the reported information, appropriate action is taken. Monitoring a security terrain could be an example where the sensor networks can be rapidly deployed. The sensors' task is to detect and to track any vehicle that could enter in that terrain. Based on the information given by the sensor, the central base unit could decide which action is better to be performed. But this action has to be taken when the information is still valid. As a consequence the communication delay has to be as short as possible and more precisely all pieces of information should have a limited delay which means a deadline.

The control made by the central base unit could be very slow for some kind of applications. Thus, actors (robots) are used to fulfill the actions; the central base unit only monitors the sensors and the actors. Actors are devices that are very mobile and are richer in resources that the sensors. Depending on the actors locations, the communication between sensors and actors could be done directly (if the actors are in the sensors neighborhood) or via sink. When the actors are not in the sensors radio range radius the communication between sensors and actors are made via sink (sinks). The information gathered by the actors (or by the sink) has to be reported to the central base unit for coordination purpose.

When the information arrives at the actors, they have to coordinate between them in order to take the appropriate decision based on the received information. They accomplish a specific task, demanded by the application. When actors fulfill the task, the gathered information has also to be valid. For a successful action, communication delays between sensors and actors should be as short as possible. The actors' task will modify the monitored phenomenon and thus the sensors will send new information. This chain is a real-time distributed system. It is real-time because the task accomplished by the actors has to be consistent in accordance with the physical phenomenon and the task has to be conforming to the application demand. The task made by the actors has a window of time to be fulfilled unless the task is useless. This window of time is determined by the validity of the information gathered by the sensors. The application will fail if the actors are late when they accomplish the task. When the actors are late, the monitored phenomenon has already changed and the actors' action is useless. These are real-time applications. As a consequence, in order to keep the information valid, the application has to define a deadline for the data units that comes from the sensors to actors and contains information. All data unit communication delays should be less than the deadline.

• Passive mobility: The sensors do not have mechanical elements to move, but some of them could be moved in an unattended way by wind or other nature phenomena. A sensor with a finished power supply disappears from the network. These facts could be regarded as sensors mobility but it is a passive mobility. The direct consequence of the passive mobility is that the sensor network topology is changing. The routing protocols have to be tolerant at these changes.

• Data-centric: the data and the geographical location from which the data is gathered are more important than the identity of the sensor. The identity of the sensor has a local importance. In the sensor network are important: the gathered information puts and the area where this information comes from. This is different than traditional networks which are address-centric, where, (for routing protocols) the global address is more important than the data conveyed. The data-centric characteristic demands new routing data-centric protocols.

• Resources constraint. A sensor is a device with little resources in memory, central processor unit (CPU) and energy. The power supply conservation for sensors has to be taken into consideration. One third of the power energy on a sensor is used for communication purpose [13]. As a consequence, the communication protocols must be designed as simple as possible in order to use less memory, less CPU and low energy consumption.

• Distributed system: The sensor system and the actor system interact in order to achieve the goal of the application. The actors have to coordinate between them to take the most appropriate actions. The sensors have to exchange location and time information between them in order to maintain the network.

• Unreliable connectivity: The connectivity between sensors could be lost because the unfriendly deployed area (interferences) or because the wireless communication system. This makes the sensor networks unreliable thus the protocols have to be robust and efficient.

• Sensor distribution: The sensors could be deployed over a large area (over 10 km diameter) and the radio range of the sensor communication system is about of tens of meters which make the density distribution an important parameter. The density parameter will be taken into consideration when we will analyze and will make the model of the delay communication between sensors in the third chapter. The sensor distribution has also another aspect: the distribution uniformity. As the sensors are deployed over a large scale, deployment could be non uniform. The direct consequence of this aspect is that protocols have to fill holes to maintain the network connectivity.

• Sensor network and actor network: The sensors are devices with limited resources and are different from the actors that are devices with rich resources. The actors have to be very mobile to make quick and accurate actions. The sensors have a passive mobility. As consequence we regard the sensor and actor networks as two different networks. The characteristics of the actor network make it a wireless ad-hoc network and could utilize all the communication protocols designed for MANET networks. The sensor network is a different wireless network because of its mainly limited resources and passive mobility. As a consequence, the research made for the ah-hoc network cannot be entirely applied to sensor network. New or adapted protocols have to be designed for this network.

2.1.2 Real-Time Communication Types

In the classical real-time applications, the same system gathers the information from integrated sensors and makes the actions using acting devices also integrated in the system. For the sensor-actor applications, the system is distributed between the three elements: the sensors which sense the environment, the actors which take the appropriate actions and the communication system that conveys the information between sensors and actors. As a consequence, the characteristic of a real-time application will demand a real-time communication.

We have to differentiate between two types of real-time communication.

As shown in the "Real-Time Systems" [14], the real-time communication refers to the traffic "*which require some degree of guarantee for time delivery*" and the application "*is not seriously affected by the end-to-end delay suffered by the packets.... In contrast, delay jitter and throughput are important*". For these applications (usually audio and video application) a transport protocol is built: Real-time Transport Protocol – RTC. This is one type of real-time communication: the transport time is important but not critical, the jitter delay of the transported data is critical. The receiver uses a buffering mechanism to eliminate the jitter delay of the data units that are delivered at the user.

The second type of real-communication is a communication where the information carried by the data units has a limited validity thus the transport time is critical. This is the case of sensor networks applications. The data units have a limited interval of time in which the information is valid as the measured phenomenon is changing. For these data units, a limited communication time is defined which is the deadline. This type of communication needs a specific transport protocol in order to respond to the application time requirements. Therefore the transport protocol that we have to design will be used for the second type of real-time communication.

2.1.3 Data Delivery Time or Travel Time

A real-time communication for sensor application implies the time necessarily to transport data units between source and destination, more precisely between the two ends of the communication. As this time is related to the sensor networks we name it also delivery time, or travel time. We name hop delay or hop time, the time for a data unit to pass from one sensor to the next sensor. As can be seen, the travel time is the sum of all hop times (hop delays) of hops along the path through the sensor network.

The objective of this work is to build a transport protocol that could be utilized at the sensor networks applications. The work concerns only the data unit delivery time between sender transport entity and the destination transport entity. At the sender the time delivery starts when the data unit is received by the transport entity – $\mathbf{t_{start}}$ in the figure 2.2. At the destination, the time delivery ends when the data unit quit the transport entity and is handled to the application entity – $\mathbf{t_{arrival}}$ in the figure 2.2. In every transit node, the data unit will be processed and relayed to pass toward the next sensor, up to the destination. In order to multiply the possible sensor networks application the transport deadline data units should be as little as possible.

The data units that come from the sink or actors and have the sensors as destinations do not have a deadline to be delivered. More important is that these data units should be reliable because they are queries for sensors. As can be seen the transport protocol should not be a symmetric protocol.

The source node is any sensor that generates a data unit which carries the information about the physical phenomenon requested by the applications. Any sensor in the network could be a source for data units. Only at source sensors the data units that convey the information pass from the application entity towards the transport entity. Thus the time delivery for the data units starts only at the source sensors.

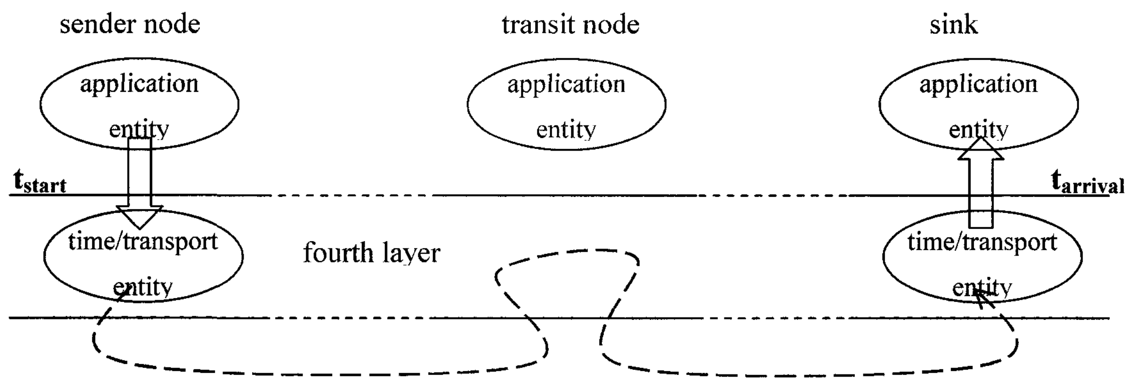


Figure 2.1 Data delivery time – the protocol objective

The transit node is the node where the data units that came from other nodes are processed and forwarded toward destination. Any sensor in the network could be a transit node.

The destination node could be the sink or an actor. When the data unit that carries the information arrives at the sink or at an actor and is handled to the application entity the delivery time stops. The sink is a gateway between the sensor network and the actors or between the sensor network and the base center unit. The communication part between the sink and the actors or the base center unit does not concern this work.

The sink (sinks) network is not mobile or it can be said it has a passive mobility as we describe the characteristic of the sensors in paragraph 2.1.1 "Characteristics of Sensor Networks". The actors are mobile and their moves are made in the proximity of the sensors. For the real sensor networks it is advisable that both sinks and actors should be on the field. In the case when all actors could overpass the radio range radius of the sensors, the communication between the sensors and actors could be lost. Thus the sink (sinks) is necessary to relay the data units between sensors and actors.

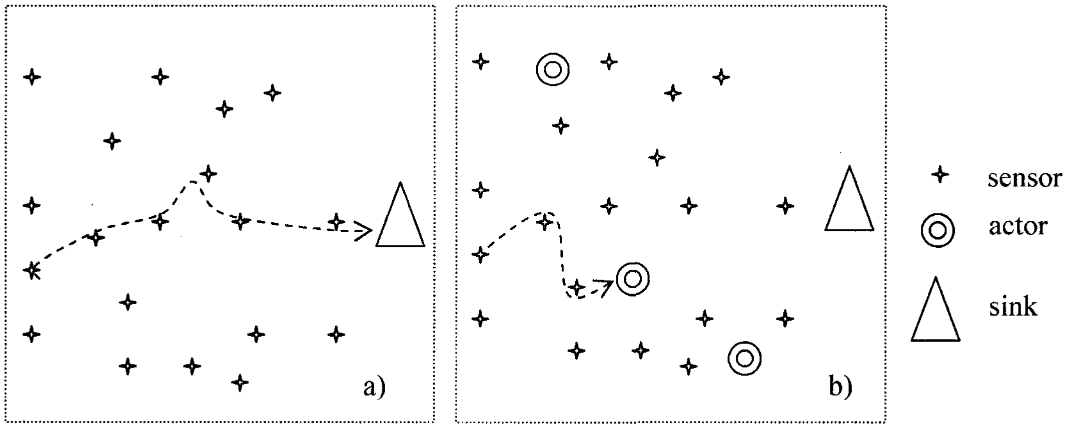


Figure 2.2 The mobile actors could decrease the message hop number

When there are not actors between sensors, the data units have to cross all the sensor networks. This is the most disadvantageous scenarios for delivery time as is traced in the figure 2.2.a. When the actors move between sensors, the actors can receive the data units (these actors become a destination) from the sensors that are in the vicinity and thus they decrease the number of hops. As the number of hops decreases, the delivery time decreases (figure 2.2 b). We take into consideration the longest delivery time: when the data message has to travel between a sensor and the sink. As consequence the destination for the transport protocol will be considered the sink (sinks).

2.1.4 Performance Metric

A data unit with a transport time greater than the deadline is a data unit that missed the deadline, while a data unit with a transport time less than the deadline is a data unit that met the deadline. In the literature a performance parameter for a real-time system communication is the percentage of data units that miss their end-to-end deadlines over all transported data units, called "*Packet Deadline Miss Ratio*"- PDMR

In the fourth chapter where the transport protocol is evaluated, we are making a comparison between the transport protocols using other performance metrics. The designed protocol in this thesis – Bonaventura (BVP) – will be compared with UDP.

Two performance metrics will be used: the data units travel time and the timelyput. The timelyput is the throughput of the data units that arrive in time at the application layer of the destination node. This metric will be better defined in the paragraph: 4.3.5 “Timelyput Definition”. We consider that these performance metrics are better than packet deadline miss ratio for real-time communication evaluation.

2.2 Layer Two Achievements in Real Time Communications

Layer two should support real-time requirement as it has to achieve efficient communication between two adjacent machines. In the literature, in order to obtain a minimum delay per hop, the data-link protocol has to solve two problems. First it has to know the sending order of the frames toward the neighbors. It is obvious that the frames cannot be sent in the same order that the node receive them from the network layer. The frames have to be ordered taking into account their time margin up to the data deadline. Second, the node has to avoid the collisions and therefore it is necessary a schedule mechanism, between the nodes that belongs to the same neighborhood, to seize the medium in a fair manner. When a collision appears two nodes have to wait until a new transmission is possible. This will increase the network latency. Usually, the wireless communication system in sensor networks uses a single channel (no spatial division) with an average radio range of tens of meters (40-100m). The transmission rate is usually 64-2048 kbps.

There are three different classes that can be analyzed regarding to the time requirement [1]. This taxonomy refers to the modality of the medium seizing in order to avoid the collisions: 1. Time scheduling, 2. Contention-based and 3. Collision-free Real-Time scheduling.

For the time scheduling class, the time is divided in slots and a scheduling algorithm determines the slot where a node can transmit. That slot in assigned to a specific node, and thus the collisions are avoided. This is the case with time division multiple access (TDMA) scheduling. Using this algorithm it is possible to avoid collisions, it minimizes the packet latency and it uses the channel in a fair way. But the proposed protocols require global connectivity information and excellent clock synchronization. This is very difficult to maintain in a topology as in sensor networks.

For the contention-based class, the protocols are based on the carrier sensing and collision avoidance mechanism. This is the case of the protocol for the standard IEEE 802.11 where a distributed coordination function (DCF) is used for basic access method. DCF uses request to send/clear to send (RTS/CTS) messages to avoid the collisions. This method cannot be used in sensor network because it does not provide any real-time guarantee thus it will be difficult to adapt this mechanism to a real-time requirement.

2.2.1 Collision-Free Real-Time Scheduling

The class Collision-free real-time scheduling uses the algorithm Earliest Deadline First (EDF) to schedule the seizing of the medium without collisions. This is the most interesting method regarding the real-time requirements. The schedule is built using this metric: the shorter deadline of the message, the higher priority will be assigned, the sooner the message will be transmitted. This is the base of the mechanism called Earliest Deadline First –EDF. This scheduling method can be used for a real-time communication because it takes into consideration the deadline of the data unit.

Two protocols were chosen using EDF to be briefly presented: the first one imposes that the network be divided into cells, the second imposes that entire network is fully connected.

An Implicit Prioritized Access Protocol for Wireless Sensor Networks

This article, presented by Caccamo et al. [2], proposes a protocol which exploits a fundamental assumption: the periodic nature of the message sent by the sensors toward the actors (more precisely the majority part of the messages is periodic and a little part is non-periodic). Also, it is assumed that the nodes are fixed.

In the layer two, the most important aspect is to know the node that needs the wireless medium at a given time. The periodic nature of the traffic gives the possibility to define an implicit deadline time for every message.

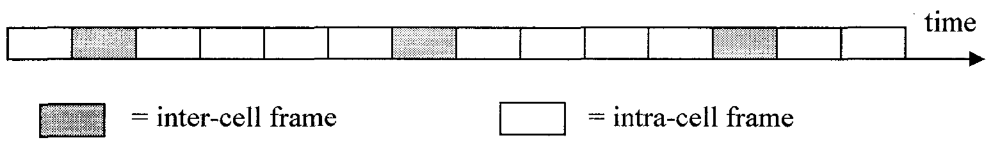


Figure 2.3 Periodic patterns for inter-cell frame transmission

The time is divided into frames (time intervals) which are used for inter-cell or intra-cell communication. The inter-cell frames appear periodically according to a predetermined pattern (figure 2.3). All nodes are synchronized on frame. In order to be a scalable protocol for a large deployment of the sensor network, the authors have adopted a cellular structure for the network architecture: The entire network is cut in hexagonal cells, using at least 7 channels (Frequency Division Multiplexing – FDM is used to communicate). In each cell the nodes are fully connected and the communication between them is multicast. A node, usually in the center of the cell, will have the function of "router". The router has two transceivers to transmit and to receive in the same frame. A message for intra-cell communication is sent first to the router during an intra-cell frame and the router will send it to the next router in an adjacent cell during an inter-cell frame.

A message has the following attributes: i) the length that is the number of frames required to be send, ii) the period time that is the number of frames between two consecutive sampling (this period is also the message deadline) and iii) the node identity.

The intra-cell messages are scheduled based the EDF algorithm: the first node to use the wireless medium for transmission is the node having a message with the shortest deadline. For the same deadline time between two nodes, the node with the higher rank will transmit the first (a rank is assigned to the node identity).

The inter-cell communication appears during inter-cell frames (figure 2.3). These frames have to be synchronized between cells and therefore all the network has to be synchronized, which is a disadvantage for this protocol. During an inter-cell frame the router will transmit using the channel of the cell to which it belongs. The router will receive on the channel assigned to the cell from which the message comes. At the router, the messages to be transmitted are sorted based also on EDF algorithm. Using a hexagonal form of the cell, transmission and reception for a router is made on a predetermined direction: there are six possible directions that are assigned statically.

The EDF mechanism provides a contention free nature, a better throughput for the network and takes into account the deadline for the messages. The drawbacks of the protocol are: a frame synchronized network which is difficult to obtain, a complex wireless system communication for a sensor (more than one channel), a complex network architecture which has to be maintained, and also a single point of failure in a cell: the router therefore a mechanism is needed to elect another router in case of router failure.

A Robust Implicit Access Protocol for Real-Time Wireless Collaboration

This article presented by Crenshaw L. et al. [3], proposes a protocol which also uses the EDF algorithm to obtain a schedule for seizing the wireless medium based on the assumption of the natural periodicity of the sensor messages. In this sense this protocol is like the above protocol. But there are some differences. The most important is that this protocol does not require the clock synchronization for the entire network, dropping the division of the time in frames. As a consequence, is necessary a full-connection of all sensors in the network, which makes it a non-scalable protocol.

There are also two improvements for this protocol comparable with the previous one. The first one is the recuperation of the unused bandwidth. If a node finishes its transmission earlier (because the message was shorter, the time, up to the end of the normal transmission, is called budget. This budget is given to the next node but this node cannot use it to transmit periodic messages, thus making a schedule shift. The next node will transmit non-periodic messages or empty messages, up to the time for sending periodic messages. The second improvement is that this protocol ensures the recovery of the network in the presence of node failure. Each node performs carrier sense of the medium. If the medium is idle more than an established duration, it is assumed that a node that has to transmit failed and a recovery must be performed.

Although this protocol utilizes the EDF mechanism to ensure a real-time communication, it does not require a synchronized network and has two improvements over the previous protocol namely recovery and bandwidth budget, it has serious drawbacks like the necessity of the full connection of nodes which implies a long radio range (difficult to obtain for a sensor) and the non-scalability for large networks. Therefore it can be used only on little scale networks with the diameter of the sensor radio range.

2.3 Routing Protocols for Sensor Networks

The network layer of the sensor networks has to deal with such problems as follows. First is the problem of global addresses. It is not possible to use the classical global addressing scheme. As the sensor identity is not important and has only a local significance, the geographic localization gives the logical address for network layer. This implies that the sensor is aware about its geographic localization and a mechanism should perform this task. The second problem is about the avoidance of the congestion in the network. In order that the packet be delivered with the lowest delay, the avoidance of the congestion must be very restrictive. The routing protocols should be adaptive to control and avoid the congestion in the network and also the protocols should avoid the congestion zone. The third problem and most difficult is to ensure that the time delivery between source and destination will be less than the deadline.

The routing protocols developed for the wireless ad-hoc networks cannot be applied directly to the sensor networks because of their main differences: network addresses, multiple data sources, data redundancies and the resources limitations. Many routing protocols have been proposed and the routing mechanisms have taken into consideration the application oriented characteristics. But there are few routing protocols that could take into consideration the source-destination delay.

The presented classification of the routing protocols is suitable for the sensor networks [4]. In summary it will be presented the routing protocols that match better the nature of sensor networks and those that could guarantee, with some adaptations, a bounded delay.

2.3.1 Data-Centric Routing

For a data centric protocol the routed data are more important than the identity of the sensor; which it is not a global identifier. The lack of global identifiers for each node makes difficult to query a specific node. Usually the queries are related to a region in the sensor network deployment. This characteristic is different than in address centric networks where the global addresses are more important in routing than the routed data.

In the sensor network, transmitted information on the monitored area are redundant thus energy consuming. The data-centric protocols are able to make some data aggregation thus reducing the information redundancy. Another important characteristic is that the data-diffusion protocol could be adapted to a real-time communication as will be shown in the next paragraph.

Directed Diffusion for Wireless Sensor Networking

This protocol [5] (presented by Intanagonwiwat C. et al.) is based on the diffusion of queries towards the nodes of a region, collecting information that match the data queries attributes and sending back along the reverse path of queries propagation. The reverse paths are established using gradients. This protocol is data-centric because it does not specify the node address or identity and because any node that can gather information about the event specified in the query, can send back a message. The intermediate node can aggregate data, sending a more accurately report. The direct diffusion protocol consists on five elements:

1. data is named using attributes. These attributes receive values when specific queries or reports are sent. Interval and duration are two important attributes that define the period for collecting data and the duration of this task.

2. the interest is the task for sensors and is related to a specific event. The interest is diffused throughout the network.

3. the gradients set up: when the interest passes through the network, each sensor will open an entry in the table tasks with all attributes and gradients field.

4. messages with the sensor information about the events that match the interest

5. path which is reinforced by the network; the routed messages will follow this route. A node can reinforce a path by choosing only one of the neighbors (between the neighbors that have gradient), using a specified rule. As the authors suggest a path can be selected based on lower delay. This adaptation will give the opportunity to use the Direct Diffusion protocol in a real-time communication.

One advantage of this protocol is the fact that it uses a natural characteristic of the sensor network: data-centric. Another is that it can be adapted to make a delay aware routing. Direct diffusion is a robust protocol: it could establish balanced multiple paths which remove a possible single point of failure.

The protocol previews data aggregation along the return path. At the first glance the data-aggregation could be very attractive because all the redundant data along the reverse path could be removed with implications at the energy conservation and congestion avoidance. This fact will make the sensors report more accurately. In the article: "The Impact of Data Aggregation in Wireless Sensor Networks" [6] the authors analysis also the aggregation delay and show that: "*the aggregation latency could not be non-negligible and should be taken into consideration*". This aspect is very important when the protocol objective is the decreasing of the data delivery time for a real-time communication. Therefore the data aggregation along the reverse path becomes a disadvantage for a protocol that takes into account the message delay.

2.3.2 Location Routing

Location routing is also called geographic routing and the protocols that form this category make routing decisions based on the geographical distance between source and destination nodes (which is a direct consequence of the lack of addressing schema in sensor network). The location information could be utilized in routing to save the sensors energy: the query is sent only toward the interesting region, limiting the number of transmissions.

Greedy Perimeter Stateless Routing for Wireless Networks (GPSR)

This protocol (presented by Kark B. et al.) [7] describes a geographic routing protocol. The most important requirement for this protocol is that all mobile nodes have to know their geographical location which became global addresses. The originating node marks the packets with the destination location. The first node can forward the packet on greedy choice, knowing only its radio neighbors' locations. The neighbor geographically closest to destination will receive the packet. This forwarding method follows until the destination is reached. A beacon algorithm is necessary to maintain the neighbors' table with their positions. When the packet arrives at a node which is closer to the destination than all its neighbors (no better node) the greedy forwarding fails. This happens when the density of the sensors in that zone is weak. The zone is called hole or void zone. In this case the node use perimeter forwarding based on the Right-Hand-Rule: when the node receives the packet from a neighbor (the sender), the next neighbor to forward the packet is its first neighbor counterclockwise against the sender.

The advantages of this protocol are: it uses a natural characteristic of the sensor network - geographical location, it uses a little routing database because it requires propagation of topology information for only the nodes in the radio range and the upgrading of the routing table is nearly stateless (do not require any context or memory of previous updating). The major disadvantage is that the protocol does not take into account the packet delay for a real-time communication, but this routing mechanism could be improved to be delay aware.

2.3.3 Delay Aware Routing

This category includes routing approaches that are based on delay in order to meet the real-time requirements. As the location is an important characteristic of the sensor networks, the routing protocols are based on a spatiotemporal mechanism.

A Spatiotemporal Communication Protocol for Wireless Sensor Networks – Speed

This protocol (presented by He Tian et al.) [8] takes into account the delivery time of the packet. The routing decisions are based on geographic location combined with the next hop forwarding speed of the packed so that end-to-end delay is almost proportional to the distance between the source and destination. The protocol assumes that each sensor knows its geographical location. The protocol also provides a computation of the next hop forwarding speed, based on an estimation of the forwarding delay. The computed speed, for the possible forwarding nodes, has to be better than a certain desired speed. The authors do not guarantee an end-to-end delivery delay bound necessary for real-time communication but they guarantee a delay per distance unit – speed guarantee (actually inverse of the speed). As a consequence, they could guarantee the end-to end delay under a given distance which is called soft real-time communication.

For each neighbor node there are two major pieces of information: position and delay estimation for forwarding. The delay estimation is made for a single hop. It is considered as a better metric to approximate the load node than average queue size. The current delay estimation is computed using three previous last delays using the exponential weighted moving average formula. A delay is the difference between the departure moment of the packet (last bit of the packet is sent out) from the node and the arrival moment (entering in the network output queue) at the next node. The propagation delay is not considered. The current delay estimation is used for speed estimation and is notified to the neighbors.

The Speed protocol uses location (that is natural in the wireless sensor networks) and ensures a guaranteed end-to end delay under a given distance therefore for a given topology the end-to end delay could be known. The delay bound is equal to the distance between source and destination divided by the desired speed (S_{set}). It realizes a good load balancing thus a robust protocol and it has a better packet delay miss ratio (the percentage of packets that do not meet their source destination deadline) than dynamic source routing (DSR) or ad-hoc on-demand vector routing (AODV) protocols. But the disadvantage of this protocol consists in the fact that it is a complex protocol that is not convenient for sensor networks.

There were presented some important routing mechanisms and some examples of routing protocols for the sensor networks. Except for the last class, the other classes could be improved with a delay-aware metric in order to be used for real-time applications.

2.4 Transport Layer

At the transport layer, there are two major problems for the sensor networks: data reliability and the timeliness for end-to-end delivered data (for real-time applications).The information gathered by sensors is redundant and could be related to the same physical phenomenon. The redundant data increase the reliability probability of the delivered data but reduce the consistent information network throughput. The redundant data will allow some losses in the network and the communication reliability remains unchanged but the redundant data can raise the risk of congestion and therefore it increases the communication delay. On the other hand the coordination between nodes for more accurate data and the aggregation of data along the path takes time. These actions will decrease the time margin of data to remain valid. The sensor networks characteristics make difficult the use of the developed transport protocols. The performances of the developed protocols are counterbalanced by the sensor network features.

2.4.1 Transport Protocols – Spatial Protocols

The transport protocol makes the link between the upper levels of the communication model and the lower layers. Its goal is to transport the data unit from one end to the other as the application layer demands. The main difference between transport and network protocols is that the transport entities exist and work only at source and at destination and not in every relay along the path as the network protocols: the transport protocol is concerned only with the end-to end data convey. In all descriptions about the transport protocols the spatial idea is always reinforced. There is no concern with the time connected with the transported data from source to destination. Therefore we can say that the transport protocols are spatial protocols, they works to convey data from one end to the other end.

In the following paragraphs we evaluate the spatial orientation of the TCP and

UDP protocols, some possible time aware mechanisms and a possible solution for WSN.

TCP is an excellent transport protocol designed for wire lines networks. It is designed to transport chunks of data in a reliable way, but without a delay guarantee. TCP, is complex and has a mechanism to correct data, using retransmission. It also has mechanisms to control the flow and congestion. These mechanisms give to TCP its universality. The flow control ensures that source and destination could exchange the segments even if the destination is slower than source. The congestion control ensures the transport over networks that are slower than other networks along the path. All these make the TCP a universal protocol: any two equipments could transfer data using any network. But for the TCP authors the transportation time was not so important. The time in this protocol was taken into account only to avoid deadlocks in the protocol algorithms. As a conclusion, TCP is a universal spatial protocol that ensures a reliable transportation of data chunks. TCP was not designed for wireless networks and in particular for sensor networks. The real-time nature of the sensor information makes the TCP inappropriate to be used in the sensor network. The TCP performance degrades significantly in wireless network because of its mechanisms that are inappropriate for this kind of network.

UDP is also a protocol for transport which is usually used for the dialog client-server: “question-answer”. Its entities work as TCP entities, at the two ends of the communication. UDP is very simple: it does not have control mechanisms. It does not guarantee the reliability of the data and it does not make the flow and congestion controls. As a simple protocol, UDP does not utilize too much system resources. As a consequence, it is better than TCP for the wireless networks. UDP, as TCP, does not guarantee the end-to-end delay of transported data. As a consequence, UDP is a simple universal spatial protocol. It has not an explicit mechanism to control the delay time of the data, thus it does not guarantee a limited delivery time for data. UDP could be used in sensor networks but it would be inappropriate for the real-time applications of these networks.

2.4.2 Simple Real-Time Mechanisms

We have chosen to evaluate transport protocols with two simple real-time mechanisms: priority scheduling and filtering.

For non-real-time communication, the data units are sent in the order in which they are processed. This sending order is called scheduling. The "normal" scheduling is First In First Out (FIFO) for non-real-time communication. In time aware application, each data unit has a deadline: the time to be transported from end-to-end has to be bounded. In this case FIFO scheduling does not work properly because the data units with longer deadline will disturb the data units with shorter deadline. The data units with shorter deadline will miss it. To avoid this, the sending order has to be changed: the data units with shorter deadline have to be sent before the data units having a longer deadline. In order to implement this, every data unit is evaluated based on different metrics (usually time and space) and will receive a priority. The data unit with higher priority will be scheduled before the data units with lower priority. Thus the priority ensures the order of the data units to be sent.

When a data unit is sent by the application layer, it is marked with the time stamp. The data unit will travel over the network and at the end it will be verified for miss deadline. If the deadline is missed, the data unit has lost the validity and it is useless. Thus, the data unit is discarded: filtered. If the data unit has met the deadline, it is valid and is used by the application layer.

2.4.3 Transport Protocols Adapted with Real-Time Mechanisms

We try to evaluate what happens when the two real-time mechanisms are adapted to a general transport protocol. As the transport protocol entities are implemented only at the two ends of the communication path, these mechanisms are used also only at the two ends.

General Transport Protocol

It is supposed that at the source, a data unit is prepared by the application layer and it is given to the transport entity. This entity will assign it a priority in order to be scheduled. At source it is useless to run the filtering mechanism because the data are still valid. Based on evaluated priority the data unit is scheduled and sent over the network. In all nodes along the path, the data unit will not be evaluated for priority or for validity because the real-time mechanisms are implemented only at two ends. In a transit node, the priority received at source is useless because there is no transport entity to use this priority. More over, data unit priority could change along the path (the data unit could be in scheduling competition with data units that has shorter or longer deadline than its deadline), but this changing is impossible to be made because of the missing of the transport entity along the path.

At the destination, the transport entity will run the filtering mechanism to verify if the data is still valid, and the application layer will receive only the valid data. This filtering is made only at the end of the path even if the data unit has lost its validity along the path and became useless. Thus a useless data unit will be transported up to the end of the path, which is not efficient. The priority scheduling mechanism is useless at the destination.

So, the two real-time mechanisms implemented in classical (spatial) transport protocol will not improve the time behavior of the communication. The priority scheduling mechanism will be applied at source and the filtering mechanism will be applied at the destination. The percentage of the missed data units will be very high and will depend only on the networks capability for transport time.

TCP and UDP in Real-Time Communication

If we adapt TCP with priority scheduling and filtering mechanisms, the source entity will prioritize the data unit that will be sent and the destination entity will filter the data units at the end of the path. The TCP mechanisms will make that TCP will behave worse than a general transport protocol. Setting up a connection takes time that has a negative impact for real-time transported data where time is important. The flow control and the congestion control mechanisms will slow the sender for the reasons that happen at the end of the path or somewhere along the path, which again, has a negative impact for the data. The correction mechanism by retransmission has a catastrophic consequence from the time point of view because the retransmission will force the data to travel two times over the path. In this case the data unit will miss the deadline and thus the correction mechanism by retransmission will not be used in real-time communication.

If we adapt UDP with priority scheduling and filtering mechanisms, it will behave better than TCP because of its simplicity: it will not try to retransmit the data unit, and it is not concerned about the latency of the receiver or the network congestion. The data is sent and the network will make the best effort to deliver the data at the other end. UDP is not aware of the time requirements for the data units along the path. All the observations made for a general transport protocol apply for UDP. As is behavior is better than TCP in time aware applications, UDP is used for this kind of communication.

As can be seen, it is very difficult to adapt classical transport protocols. These protocols are fundamental spatial protocols that "think" in a global manner: end-to-end. The two shown real-time mechanisms could not balance the spatial behavior of the transport protocols.

Chapter 3 Protocol Design

In this chapter we design the transport protocol named Bonaventura. The chapter has four subchapters as follow: in the first subchapter we justify the fundamental idea to put a time entity in every node along the data route. The second subchapter deals with geometrical considerations in the sensor network. The third subchapter is dedicated to the model, based on which we can propose the protocol algorithms. In the last subchapter are described the the protocol mechanisms and a possible implementation of the mechanisms.

3.1 Local Time Transport Entity

As it was demonstrated, the design goals for a transport protocol are very different from real-time communication goals which consequently they cannot serve. The essence of a transport protocol is that it has a global behavior, it regards only the result made by the lower layers at the ends points and try to correct them but these corrections are not time aware. This behavior is given by the fact that the transport protocol entities run only at source and destination places which give the possibility to name this design type as a spatial transport protocol.

3.1.1 Time Transport Protocol

It will be interesting to make an essential modification at the transport layer: obtain global (end-to-end) good results using a local control (at each step) and make the imposed corrections locally. At every network node it is possible to implement the time mechanisms on the fourth layer, thus the transport layer will became a time control layer that will manage the time requirements in each node along the path between source and destination. The figure 3.1 gives the new communication layers model.

This model is suited for the sensor network where on each node exists all stack layers from application to the physical.

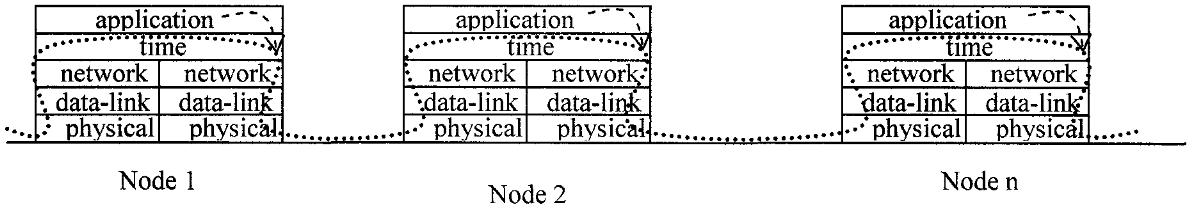


Figure 3.1 Layers model and the data unit travel in the real-time communication

In proposed architectural layers, in a node, the application layer will handle the data unit received from the sensor system to the transport layer. Here, the transport entity will evaluate it against the time requirements. When the data unit arrive at the next hop, it will be brought up the forth layer where the time transport entity runs. This entity evaluates the data unit against time requirements. If the data unit is still valid, the time transport entity will calculate the priority to be scheduled based on priority scheduling mechanism. Then the data unit will be sent by the node using the routing and data link mechanisms. Therefore all the data units in that node will be transmitted based on time-requirements.

The local implemetation of the time mechanisms (in every node along the path) will improve the time behavior of the end-to-end communication. At every node along the path, the transmission order of the data units will be changed - based on their time budgets (the difference between the deadline and the elapsed time). This will help the data units to be in time at the end of the path. If a data unit is no more valid, it will be discarded, relieving the network to carry a useless data and thus avoiding easier the congestion in the network. It will be seen that, based on the sensor network model, other mechanisms could be locally implemented in order to help the data units to meet their deadlines.

A disadvantage of this design could be that the time control at every node will imply a supplementary computation therefore a supplementary latency for the end-to-end communication. This is true, but this disadvantage is disputable in two ways: a) this time control has to be made (if not we remain at a spatial transport protocol design), and b) the time wastes at this layer could be quantized: a bounded delay.

3.1.2 The Position of the Time Entity

The place of the time transport entity on the layer stack is at the forth layer. It cannot work at another layer because:

1. using the priority mechanism between the locally generated data units and the data units in transit in a node force the time management to work under the application layer

2. to route a data unit over the network is possible after applying the priority scheduling mechanism because data units has to be routed in the order that ensure the advantage for the data units with higher priority. After priority scheduling mechanism the data could be forwarded; this forces the time entity to work above the network layer

Thus, under the application layer and above the network layer means transport layer in the TCP/IP Reference Model. This is the place for the time entity.

The time entity of the transport protocol is implemented in every sensor as it was stated in the paragraph 3.1.1: "Time Transport Protocol". The position of the time entity in the protocols stack is shown in the figure 3.2.

Every sensor in the network will generate data units that have to be reported to the actors or central base unit passing through the sinks. At the same time, every sensor will be a relay for the data units generated by other sensors. Usually, in the networking world, a relay has only three layers – up to the network layer. The data unit that arrives at a relay will pass from the physical layer toward network layer. The network layer takes the forwarding decision based on the network destination address and routing table. The data unit will pass from the network layer toward the physical layer in order to be transmitted. So, in this case the relaying is made at the network layer level.

As we proposed, for Bonaventura protocol, in the sensor network the relaying for the data units that are in transit will be made at the transport layer. The data units that arrive at a sensor will pass from the physical layer toward the transport layer. If the data unit is addressed to that sensor the time entity will deliver it at the application layer. If the data unit is not addressed to that sensor, the time entity will relay it. The time entity will apply for this data unit the time mechanisms. Here, at this level, the transit data unit will meet the data unit generated by the sensor (figure 3.2). The generated data unit will be submitted at the same time mechanisms in the time entity at the transport level. Both generated and transit data unit will pass from the transport layer toward the physical layer in order to be transmitted to the next sensor. Therefore the generated (data units that inform about the monitoring phenomenon) and transit data units will be treated in the same way by the time entity.

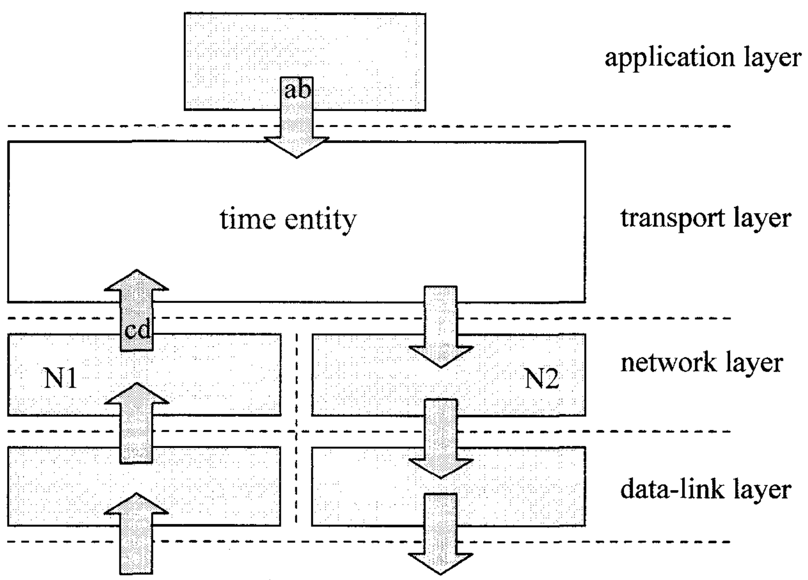


Figure 3.2 Position of the time entity in the protocols stack for every sensor

As every node in sensor network generates the information about the environment, each node will have the entire protocol stack, and therefore the time transport entity can be easily implemented on the forth layer. Thus, from the communication point of view, the sensors will be identical and the network homogeneous.

On the other hand, the wireless communication that usually has to be aware of the node channel status, demands a local management of the communication that corresponds to the time management which has to be made locally. This aspect can open the way to optimize the protocols stack not only in the layered manner but also in a cross-layer design layer manner.

As a consequence, the sensor networks can be utilized to become a real-time network for time aware application. This will be developed in the following paragraphs.

3.2. Sensor Networks – Geometry Considerations

In the article [17] there is an analysis on the real-time capacity limits for a wireless sensor network. The authors give an equation of the real-time capacity of the sensor network based on the network parameters. A message has also a capacity requirement in order to be schedulable. The authors show that all the messages in the network are schedulable if the sum of their capacity requirement is less than the network real-time capacity. The authors do not take into consideration for an analysis the geometry features of the sensor network. The sensors are deployed over an area; they have an average density and a radio range radius. The results of this analysis should be different depending on these parameters.

3.2.1 Sensor Network Parameters

We will start with some important parameters and as the evaluation goes forward other parameters will be defined. All parameters are put in the Table 3.2 "Time equation parameters".

N = number of sensor in the network [nb]

r = radio range radius [m]

W = transmission rate [kbyte/s]

L = data unit length [byte]

S = number of sinks [nb]

As it was shown in the second chapter, we do not consider the situation when the actors are moving inside the sensor networks. As the moving actors improve the delivery time, we will consider the most difficult situation: the data units will be delivered to the sink nodes.

As the sensor network is a network deployed on the field and the communication between the sensors is a wireless communication, we consider another important parameter that influences the data unit travel time, the average surface density of the sensors. Intuitively, the travel time for a data unit over the sensor network is different when all the nodes are deployed on a surface covered by a single radio range or the nodes are deployed on a bigger surface that can be covered with a number of radio ranges. At the end of the analysis we will evaluate the importance of this parameter in the travel time equation.

Let note a_N the average surface density of the N sensors; the measure unit for the average density is "nodes/m^2".

3.2.2 Derived Network Parameters

The network area (A) on which the sensors are spread is:

$$A = \frac{N}{a_N} \text{ or } N = Aa_N$$

When a sensor starts a transmission, all the sensors in the radio range radius will be affected: they cannot nor transmit neither receive from another sensor. The affected area is πr^2 and all the sensors in this area have to wait until the transmission is finished. All the sensors than can hear the transmission of a sensor j is a set named *neighborhood (j)* [17]. Figure 3.3 represents the neighborhood model for a sensor j; all the sensors in the circle are affected by the sensor j transmission.

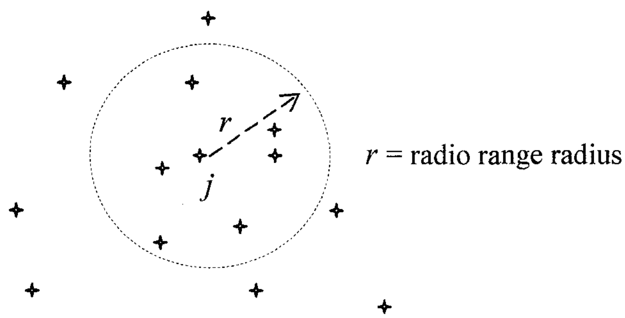


Figure 3.3 The neighborhood model for a sensor

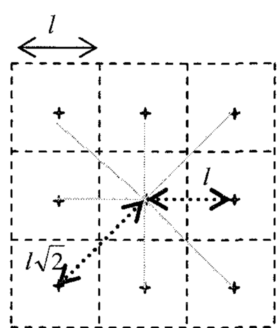
The average number in a sensor neighborhood noted with m is given by:

$$m = \lceil \pi r^2 a_N \rceil \qquad \text{Eq.3.1}$$

This is an important derived parameter for the sensor network and it can be seen as a concentration factor, because only one from m sensors can utilize the communication system; the other sensors have to wait.

Another important parameter for the sensor network is the maximum number of hops that a data unit must travel until it arrives at a sink. The number of nodes that a data unit must travel influences the time travel because every node increases the travel time.

For this we have to calculate the average distance between nodes d_N, when we know the average nodes density a_N. The average nodes density a_N indicates that the nodes are deployed in an uniform manner, each node in a square as it is shown in figure 3.4. The length side of the square is $l = \frac{1}{\sqrt{a_N}}$. In the figure 3.4 there are represented 9 sensors. The distance from the central sensor to each sensor placed on the vertical or the horizontal is the same as $l = \frac{1}{\sqrt{a_N}}$. The distance from the central sensor to the each sensor placed in corners is $l\sqrt{2}$. To calculate the average distance between the nodes we have to calculate the arithmetical mean of the distances between the central node and the surrounding nodes. Thus the average distance is $d_N = \frac{l + l\sqrt{2} + l + l\sqrt{2} + l + l\sqrt{2} + l + l\sqrt{2}}{8} = l\frac{1+\sqrt{2}}{2}$.



$$l = \frac{1}{\sqrt{a_N}}$$

Figure 3.4 Distance model between nodes

In consequence, the average distance between nodes is:

$$d_N = \frac{1+\sqrt{2}}{2}\frac{1}{\sqrt{a_N}} = \frac{1.2}{\sqrt{a_N}};$$

The number of nodes in a distance D is:

$$N_D = \left\lceil \frac{D}{d_N} \right\rceil = \left\lceil 0.83 D\sqrt{a_N} \right\rceil \qquad \text{Eq.3.2}$$

As the sensors are deployed in an unattended manner, we make the assumption that the sensor network is deployed on a surface of a disk, with the average density of the nodes of a_N *nodes/m²* (figure 3.5). The disk radius R is given by the equation:

$$\pi R^2 = A = \frac{N}{a_N} \text{ thus } R = \sqrt{\frac{N}{\pi a_N}} \qquad \text{Eq.3.3}$$

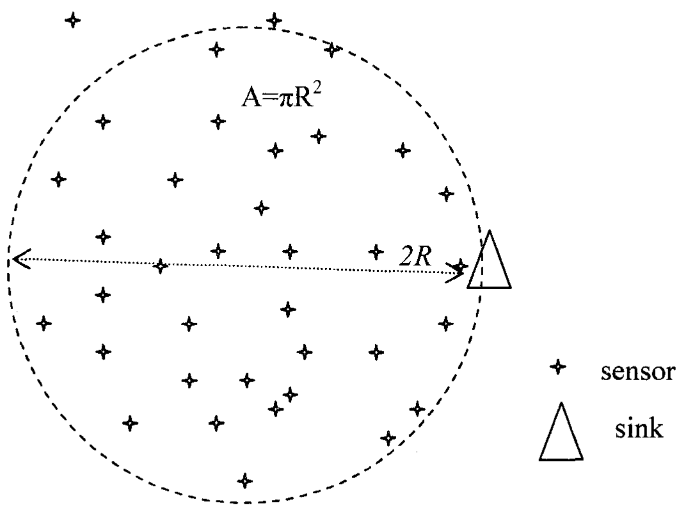


Figure 3.5 The sensors network model surface and the maximum distance in the network

As we want to calculate a limit of the travel time we consider the most disadvantaged situation. The sink is located at the edge of the disk surface sensors deployment and a sensor from the other edge sends a data unit. Thus the data unit has to travel up to the sink a distance of *2R*, which is the diameter of the disk area (figure 3.5). It is obvious that the other sensors all over the deployment area are closer to the sink that the considered distance: $D_{\max} = 2R$.

We have shown in the second chapter that the propagation time is negligible (tens of μs). Therefore we are interested of the maximum number of nodes $N_{\max}$ (the nodes introduce the delays) that a data unit has to travel up to the sink. We use the equation 3.2 and we replace the distance D with disk diameter: $D_{\max} = 2R = 2\sqrt{\dfrac{N}{\pi a_N}}$ (from equation 3.3).

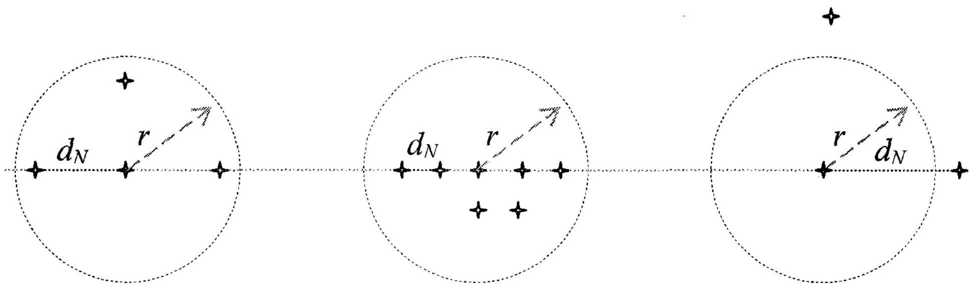
$$N_{\max} = \left\lceil 0.83 \cdot 2\sqrt{\frac{N}{\pi a_N}}\sqrt{a_N} \right\rceil = \left\lceil \frac{1.66}{\sqrt{\pi}}\sqrt{N} \right\rceil = \left\lceil 0.93\sqrt{N} \right\rceil;$$ thus we can approximate:

$$N_{\max} = \left| \sqrt{N} \right| \qquad \text{Eq.3.4}$$

✛ sensor

r = radio range

d_N = average distance between sensors



a. $\frac{r}{2} \le d_N \le r$ b. $d_N \le \frac{r}{2}$ c. $d_N > r$

Figure 3.6 The possible relation between radio range radius r and average distance between sensors d_N

This result does not take into account the radio range radius r. The result is only a geometrical approximation of the maximum number of hops that a data unit has to pass till it arrives at the sink. This result has a limited validity (figure 3.6). It is valid when the distance between nodes d_N is $\frac{r}{2} \le d_N \le r$ (figure 3.6.a.). When the distance between nodes is $d_N \le \frac{r}{2}$, then the communication can jump over a node and the equation 3.4 is no more valid (figure 3.6.b.). When $d_N > r$, the communication is impossible (figure 3.6.c.).

3.3 The Model

The delay analysis in the sensor networks follows the time analysis in a queue system and is based on the book [18]. We have adapted the computer network analysis at the sensor networks.

3.3.1 Sensor Network Delay

All the nodes in a sensor j neighborhood (the nodes that can hear the transmission made by j sensor – Figure 3.3) are affected by the transmission made by this sensor. All sensors in the neighborhood have to wait until the medium is free in order to seize the medium. The transmission rate is W byte/s. Each node in the neighborhood has a queue of data units to be transmitted, and all the neighborhood nodes data units form a virtual queue [17] that will be transmitted on a "single channel"(figure 3.7). We suppose that the time to access the medium is negligible and the model is a M/M/1 queue. M/M/1 is a standard queuing notation where the first letter represents the arrival process, the second letter represents the service process and the number represents the number of parallel servers. In our case, the arrival process and the service process (letter M) are Markovian (exponential) distribution [19]. We choose this model because is the simplest and we estimate that it models with a good approximation the concept of the virtual queue.

We note the arrival rate data unit for a sensor i by λ_i. As the arrival processes are in parallel, the arrival rate of the virtual queue λ_M for all the sensors in the neighborhood is:

$\lambda_M = \sum_{i=1}^{m} \lambda_i$ and if we take for every sensor, the average arrival rate of λ, thus:

$$\lambda_M = m\lambda \qquad \text{Eq.3.5}$$

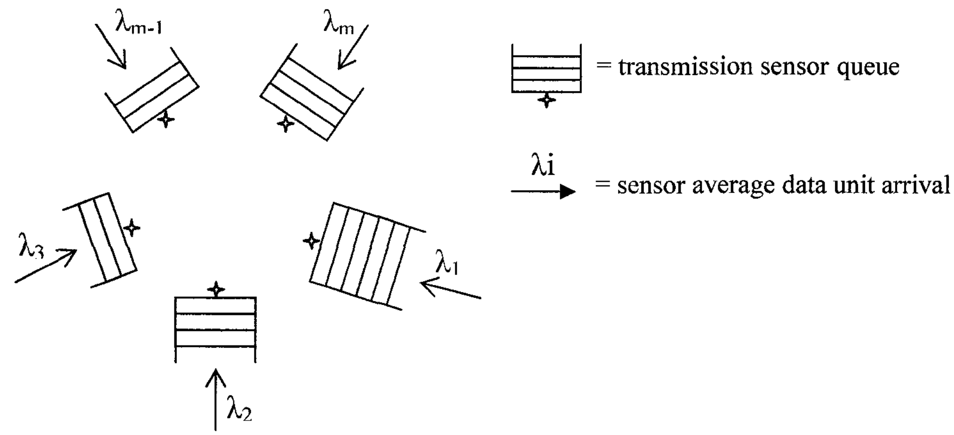


Figure 3.7 The concept of virtual queue for the sensors in a neighborhood

The data unit arrival rate represents the number of data units that arrive at the data link layer to be transmitted. It includes the data units that are generated by the sensor and the data units that are in transit (internal) in that node.

3.3.2 The Data Unit Delay

If we consider that the length of the data unit to be an exponential distribution with the average of L, then the service rate for server is $\mu = \frac{W}{L}$. The delay for a data unit per hop is [18]:

$$T_h = \frac{1}{\mu - \lambda_M} = \frac{1}{\frac{W}{L} - m\lambda}.$$

$$T_h = \frac{L}{W} \cdot \frac{1}{1 - m\lambda \dfrac{L}{W}} \qquad \text{Eq.3.6}$$

This time represents the time necessary for a data unit to travel from one sensor to another (delay per hop). This time includes the time to wait in the queue and the transmission time. It does not take into account the time that takes the sensor to seize the medium (the time to size the medium is considered zero). This delay, also, does not take into account the process time inside each sensor for every data unit.

The minimum delay for a data unit per hop is given when the average arrival rate of the data units is zero ($\lambda = 0$) [18], then:

$$T_0 = \frac{L}{W} \qquad \text{Eq.3.7}$$

The maximum average arrival rate is obtained when the delay per node is infinite. In this case the capacity of the server is equal with the flow of the data units, thus we can say that the throughput is maxim.

$$T \to \infty \text{ thus } \frac{W}{L} - m\lambda_{\max} = 0 ;\ \lambda_{\max} = \frac{W}{mL} \qquad \text{Eq.3.8}$$

We introduce equation 3.8 in the equation 3.6 and we obtain:

$$T_h(\lambda) = \frac{L}{W} \frac{1}{1 - \dfrac{\lambda}{\lambda_{\max}}} \qquad \text{Eq.3.9}$$

The graphic of this variation is represented by the figure 3.7.

If we note $\alpha = \dfrac{\lambda}{\lambda_{\max}}$ Eq.3.10

We obtain: $T_h = \frac{L}{W}\frac{1}{1-\alpha}$ or $T_h = T_0\frac{1}{1-\alpha}$ Eq.3.11

where: $T_0 = \frac{L}{W}$; $\alpha = \frac{\lambda}{\lambda_{\max}}$; $\lambda_{\max} = \frac{W}{mL}$

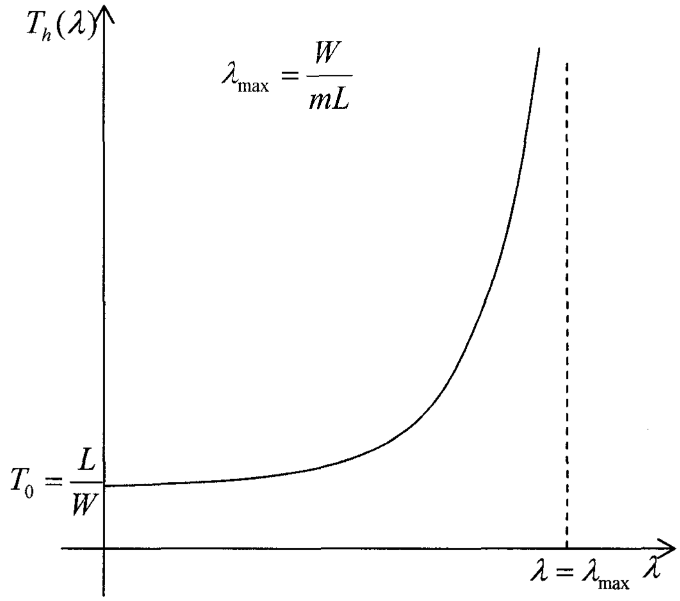


Figure 3.8 The delay variation per hop vs the average arrival rate of the data units

As it can be seen from the graphic, the time for a data unit to pass a hop is not bounded. The time increases boundless when the average arrival rate gets near the maximum average arrival rate λ_{max}. As a conclusion, in order to have a limited delay per hop we have to keep the average arrival rate at a value less than the maximum average arrival time ($\lambda < \lambda_{\max}$). Each sensor that belongs to the neighborhood (that means each sensor from the *m* sensors) could have this maximum data unit arrival rate. The maximum average arrival rate (from equation 3.8) depends on the transmission rate *W*, average data unit length *L* and the number of sensors in the neighborhood *m* which depends (from equation 3.1) on the average surface density of the sensors in the network: a_N.

If we consider also the maximum number of hops to travel a data unit over the sensor network, given by the equation 3.4, we obtain the maximum travel time (delivery time) of a data unit to reach a sink: T_M:

$$T_M = N_{\max} T_h = \left\lceil \sqrt{N} \right\rceil \frac{L}{W} \cdot \frac{1}{1 - \frac{\lambda}{\lambda_{\max}}} = \left\lceil \sqrt{N} \right\rceil \frac{L}{W} \cdot \frac{1}{1-\alpha} = \left\lceil \sqrt{N} \right\rceil \cdot T_0 \frac{1}{1-\alpha} \qquad \text{Eq.3.12}$$

This equation does not take into account the variation of the arrival rate of the data units along the path.

3.3.3 The arrival rate for data units

From the equation 3.10 and 3.8 we can write:

$\lambda = \alpha \frac{W}{mL}$ and our aim is to keep the value of α ($\alpha = \frac{\lambda}{\lambda_{\max}}$) at a constant value in order to limit the delivery time. In this case there is a dependence of the average arrival rate (λ) inversely proportional to the number of nodes in the neighborhood (*m*).

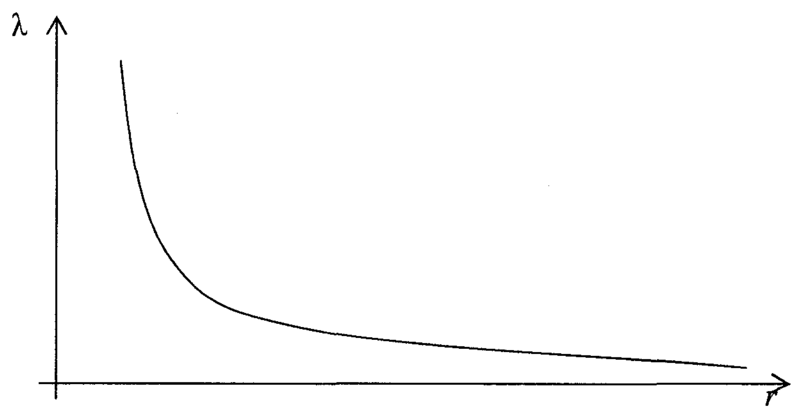


Figure 3.9 The arrival rate data unit vs radio range radius

But using the equation 3.1 that gives the dependency of the number of nodes in the neighborhood we can write:

$$\lambda(r) = \alpha \frac{W}{L} \frac{1}{\lceil \pi r^2 a_N \rceil} \qquad \text{Eq.3.13}$$

As we can see there is a dependency between the average arrival rate and the radio range radius which is shown in the figure 3.9. If we want to control the delay for the data units to pass from one sensor to the other it is possible to increase the average arrival rate by decreasing the radio range radius. If the ratio $\alpha = \frac{\lambda}{\lambda_{\max}}$ remains constant, the delay will not be increased (the consequence of the equation 3.11). The average arrival rate λ could be increased by decreasing the radio range radius. The ratio $\alpha = \frac{\lambda}{\lambda_{\max}}$ remains constant because the maximum average arrival data rate λ_{max} will be increased by decreasing the radio range radius. This is very important when in a sensor the number of the data units that has to be transmitted begins to increase (that means λ increases) and we do not want to discard them. It will be very easy to decrease the radio range radius (λ and λ_{max} will increase thus α remains constant) and all the data units can be sent. When the arrival rate will become normal, the radio range can be increased to normal.

3.3.4 The Average Generation Rate

We try to find the average data unit rate that can be generated by a sensor. The number of data units generated by the entire network must be equal with the number of data units that can be carried out by the network (conservation law). The number of data units carried out by the network depends on the number of sinks used by the network.

We note the number of sinks in the network S. We should consider that all the data units generated by all sensors in the network have to be carried out by the S sinks. The model is the same as for the calculus of the delay of the data unit (figure 3.7). All sensors in a sink neighborhood form a virtual queue M/M/1. At this moment we do not take into consideration the data units sent by the sink toward the sensor network, thus all data units in the virtual queue are the data units generated by the network. All the generated data units are separated into S virtual queue M/M/1, each served by a server with the service capacity W.

Each sensor in the network will generate data units to be reported. We assume that this process has an exponential distribution. This process has a data unit generation rate γ_i, where i represent the sensor. As the sensors in the network generate data units independently, all these processes are in parallel. Thus the average generation rate for entire network is:

$\gamma_N = \sum_{i=1}^{N} \gamma_i$ and if we take the average generation rate γ equal for every sensor:

$\gamma_N = N\gamma$, where N is the number of sensors in the network.

This average generation rate for entire network has to be carried out by S servers and thus this rate is separated in S queues. For every virtual queue it will be an average generation rate γ_S given by:

$$\gamma_S = \frac{\gamma_N}{S} = \frac{\gamma N}{S} \qquad \text{Eq.3.14}$$

For the sensors in the sink neighborhood, the average generation rate that corresponds to a single sink γ_S has to be equal with the average arrival rate of those sensors (that belong to the sink neighborhood) λ_M, given by the equation 3.5. Thus $\gamma_S = \lambda_M$ and from equation 3.5 and 3.14 we have:

$$\frac{\gamma N}{S} = m\lambda$$

The average generation rate of the data units per sensor is:

$$\gamma = \frac{mS}{N}\lambda \qquad \text{Eq.3.15}$$

The average generation rate for a sensor is directly proportional to the number of sinks in the network *S*, the number of sensors in the neighborhood *m*, and the average arrival rate of the data units to be transmitted by the sensor λ. It is inversely proportional to the number of sensors in the network *N*.

We note $k = \frac{mS}{N}$. Eq.3.16

It is like a network concentration coefficient because all the data units that have to be carried out by the sinks, look like the liquid in a funnel. The data units have to pass through the sink. If the generated data units are too many the travel time becomes to increase and it is like the increasing of the level liquid in the funnel (funnel effect). Therefore we name *k*, the funnel coefficient. It makes a linear relationship between the average generation rate data units and the average arrival rate data units of a sensor. As the figure 3.10 shows, *k* is the slope of the linear dependency $\gamma = k\lambda$. The values of the funnel coefficient could be: $0<k<1$ because the average generation rate could not exceed the average arrival rate. As a design consequence, we have: $0 < \frac{mS}{N} < 1$ (from the Eq.3.16):

$$0 < mS < N \qquad \text{Eq.3.17}$$

This equation means that the product of sink number *S* and the number of the sensor in the neighborhood *m* should not be more than the number of the sensors in the network *N*.

The maximum data unit generation rate could be obtained when we have the maximum average arrival rate given by the equation 3.8: $\gamma_{\max} = \frac{mS}{N}\lambda_{\max} = \frac{mS}{N}\frac{W}{mL}$

$$\gamma_{\max} = \frac{SW}{NL}; \qquad \text{Eq.3.18}$$

We can write also $\gamma_{\max} = k\lambda_{\max}$ as is represented in the figure 3.9.

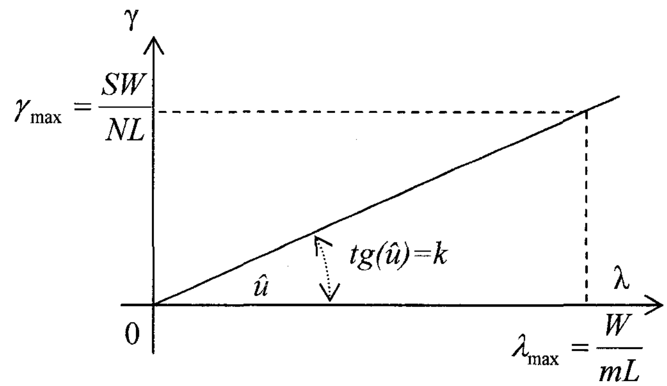


Figure 3.10 Average sensor generation rate γ vs average sensor arrival rate λ

If sensors will generate data units more than the sinks can carry out, the delay of the data units to be delivered from sensors to sinks becomes infinite, thus it is impossible to establish a bounded deadline for data units. Actually, the equation 3.18 should be a limit and must be written as $\gamma < \frac{SW}{NL}$.

3.3.5 The Average Data Units Delay and Average Travel Time

As the data units are traveling towards the sinks, the average arrival rate of the data units for the sensors is changing. The average arrival rate becomes greater and greater due the data units concentration (funnel effect). Suppose that the sinks are on the network edge, as in figure 3.5 and all the sensors in the network generate data units: The data units that come from the opposite edge (from the sinks position) toward the sink will encounter a greater arrival rate of the data units. In this case, along the path, the time spent by a data unit to pass at the next node (delay per node) increases. If we keep the same model (all the data units from the sensors in a neighborhood form a virtual queue M/M/1) we can find the average time spend by a sensor to arrive at the next sensor. At the opposite network edge of the sink, the data arrival rate is equal to the data generation rate γ (no data units in transit).

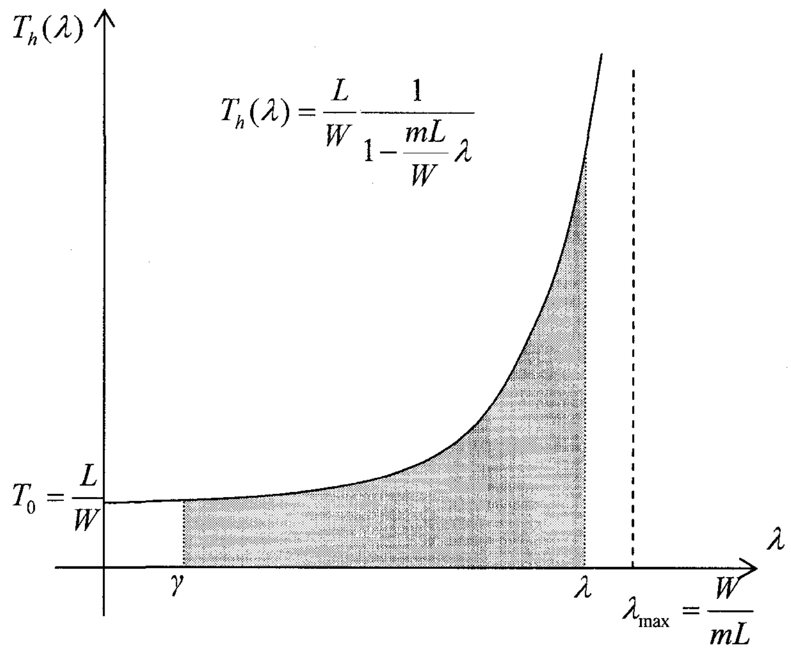


Figure 3.11 Integration limits to calculate average delay data units per hop

The data unit arrival will be greater and greater up to the sensors in the neighborhood of the sinks which will have the greatest data unit arrival rate: λ. We calculate the average delay using the integration calculus. The curve is given by the equation 3.6. In the figure 3.11 is represented the delay dependency vs the average data unit arrival rate and the integration limits: γ and λ.

$$T_{hm}(\lambda) = \frac{1}{\lambda-\gamma}\int_{\gamma}^{\lambda}\frac{L}{W}\frac{d\lambda}{1-\frac{mL}{W}\lambda} = \frac{L}{W}\frac{1}{\lambda-\gamma}\left(\frac{-1}{\frac{mL}{W}}\right)\ln\left(1-\frac{mL}{W}\lambda\right)\Big|_{\gamma}^{\lambda} = \frac{1}{\lambda-\gamma}\frac{1}{m}\ln\frac{1-\frac{mL}{W}\gamma}{1-\frac{mL}{W}\lambda}$$

We can use the relation $\gamma = k\lambda$ and the equation 3.8 for λ_{max}.

$$T_{hm}(\lambda)=\frac{1}{\lambda-k\lambda}\frac{1}{m}\ln\frac{1-\frac{\gamma}{\lambda_{\max}}}{1-\frac{\lambda}{\lambda_{\max}}}=\frac{1}{\lambda m}\frac{1}{1-k}\ln\frac{1-k\frac{\lambda}{\lambda_{\max}}}{1-\frac{\lambda}{\lambda_{\max}}} \quad \text{Eq.3.19}$$

Using equation 3.8 and 3.10 we have: $\alpha=\frac{\lambda}{\lambda_{\max}}=\frac{\lambda}{\frac{W}{mL}}=\lambda m\frac{L}{W}$ and $\frac{1}{\lambda m}=\frac{L}{W}\frac{1}{\alpha}$.

Thus:

$$T_{hm}(\alpha)=\frac{L}{W}\frac{1}{\alpha}\frac{1}{1-k}\ln\frac{1-k\alpha}{1-\alpha} \quad \text{Eq.3.20}$$

where $\alpha=\frac{\lambda}{\lambda_{\max}}$, $k=\frac{mS}{N}$, $\lambda_{\max}=\frac{W}{mL}$ and $m=\lceil \pi r^2 a_N \rceil$

The equation 3.20 represents the average delay for data units to pass from one sensor to the next sensor. The factor $\frac{1}{1-k}$, in equation 3.20 increases the delay because of the data units' concentration (funnel effect). As $k=\frac{mS}{N}$, the delay is grater when $mS \to N$.

The minimum time is obtained when the average arrival rate is zero ($\lambda = 0$) and corresponds to equation 3.7: $T_0=\frac{L}{W}$ because in equation 3.20, $\lim_{\alpha\to 0}(\frac{1}{\alpha}\ln\frac{1-k\alpha}{1-\alpha})=1-k$.

If we consider the maximum number of hops of the data unit route over the sensor network, given by the equation 3.4, we obtain the average delivery time T_m (travel time) of a data unit to get a sink:

$$T_m=N_{\max}T_{hm}=\lceil\sqrt{N}\rceil\cdot T_0\frac{1}{\alpha}\frac{1}{1-k}\ln\frac{1-k\alpha}{1-\alpha} \quad \text{Eq.3.21}$$

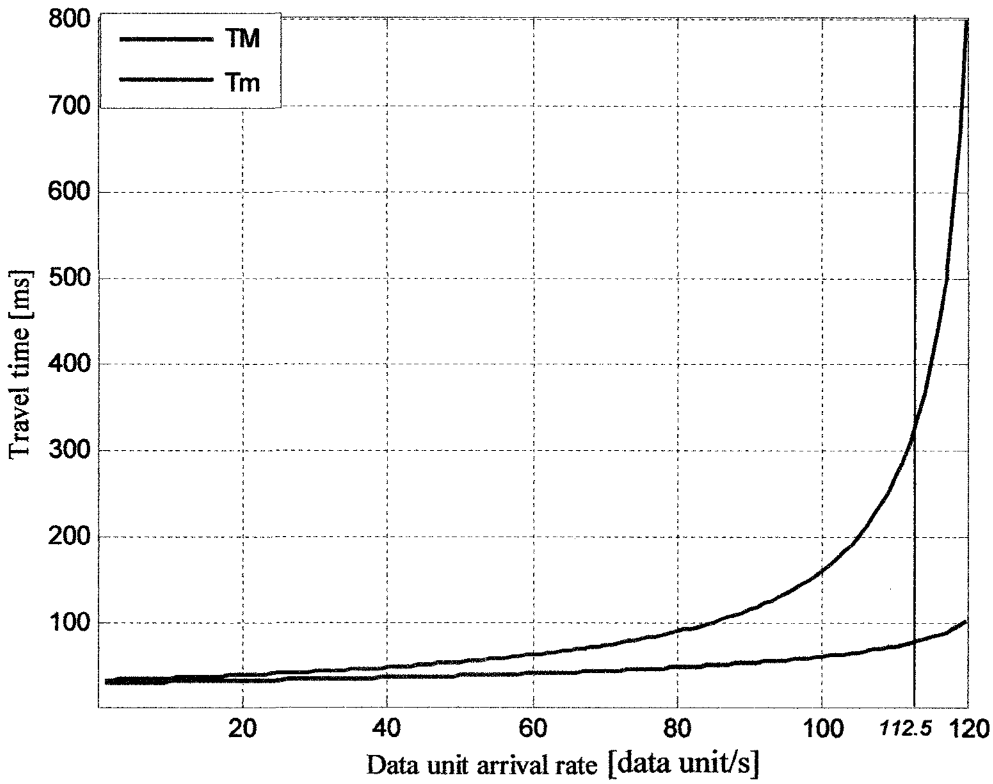


Figure 3.12 Travel time variations vs. data unit arrival rate

In the equation 3.12 we have obtained another equation for delay: $T_M(\alpha) = \lceil \sqrt{N} \rceil \cdot T_0 \frac{1}{1-\alpha}$ which is different than the equation 3.21: $T_m(\alpha) = \lceil \sqrt{N} \rceil T_0 \frac{1}{\alpha} \frac{1}{1-k} \ln \frac{1-k\alpha}{1-\alpha}$. The difference between these two equation consist in the fact that in equation 3.12 we have considered that for every step along the path, the average arrival rate is constant λ, every hop will be passed with the delay given by equation 3.11. The equation 3.21 represents delivery time per network when the hop delay is the average delay when average arrival rate is not constant but it varies from γ to λ, thus $T_m \leq T_M$ (and also $T_{mh} \leq T_h$). As a consequence, the delivery time T experimented by the data unit to pass over the network should be:

$$T_m \le T \le T_M \qquad \text{Eq.3.22}$$

If we take the folowing parameters: nodes $N = 1000$; radio range $r = 30$ m; sensor density $a_N = 25 \cdot 10^{-4}$ nodes/m^2; channel speed $W = 50$ Kbytes/s; data unit length $L = 50$ bytes; and number of sinks $S = 6$ we obtain the maximum travel time T_M and the average travel time T_m as are represented in the figure 3.11.

It can be seen that the difference between the two travel times increases as the data units arrival rate becomes closer to maximum value: λ_{max}. If the average arrival rate is $\lambda = 112.5$ data units/s then the delivery time (travel time) should be: 84.3ms<T<320ms.

3.4 The Protocol Mechanisms

The main results obtained from the model are given in the table 3.1. These results are obtained in equations 3.11, 3.12, 3.20 and 3.21 in the subchapter 3.3 "The model".

Table 3.1 Model results

Time	Hop	Network
maxim	$T_h(\alpha) = T_0 \dfrac{1}{1-\alpha}$	$T_M(\alpha) = \lceil \sqrt{N} \rceil \cdot T_0 \dfrac{1}{1-\alpha}$
average	$T_{hm}(\alpha) = T_0 \dfrac{1}{\alpha} \dfrac{1}{1-k} \ln \dfrac{1-k\alpha}{1-\alpha}$	$T_m(\alpha) = \lceil \sqrt{N} \rceil T_0 \dfrac{1}{\alpha} \dfrac{1}{1-k} \ln \dfrac{1-k\alpha}{1-\alpha}$
The meaning of these parameters is: $\alpha = \dfrac{\lambda}{\lambda_{max}}$, $k = \dfrac{mS}{N}$, $\lambda_{max} = \dfrac{W}{mL}$ and $m = \lceil \pi r^2 a_N \rceil$, The delivery time T for entire network should be: $T_m \le T \le T_M$		

The main results regarding the modeled travel time is that if we want to keep the this time limited, we should keep under control the value of the arrival rates ratio: α and as a consequence the value of the λ – average arrival rate of the data unit. So, this prefigures a mechanism to keep the value of the average arrival rate at a controlled value.

The time entity of the transport protocol for real-time communication will have therefore the two mechanisms (priority scheduling and filtering mechanisms) depicted in the 3.1 subchapter: "Local Time Transport Entity" and a mechanism that will control the value of the average arrival rate of the data units in every sensor.

3.4.1 Time Equation Parameters

The equations shown in the table 3.2 for the delivery time depends on two kinds of parameters, as follows:

a. known parameters before network deployment (named primary parameters): average data unit length L, channel transmission rate W, radio range radius r, number of sensors in the network N and number of sinks S.

b. known parameters after the network deployment (named secondary or deployment parameters): average density of nodes a_N, number of sensors in a neighborhood m, maximum number of hops in the network N_h, average data unit arrival rate λ, average data unit generation rate for a sensor γ and funnel factor k.

Table 3.2 Time equation parameters

<table>
<tr><th>primary parameters</th><th colspan="3">secondary parameters</th><th colspan="2">tertiary parameters</th></tr>
<tr><td>W</td><td rowspan="5">a_N</td><td rowspan="2">$m = \lceil \pi r^2 a_N \rceil$</td><td rowspan="2">$\lambda_{\max} = \frac{W}{mL}$</td><td rowspan="5">α</td><td rowspan="2">$\lambda = \alpha\lambda_{\max}$</td></tr>
<tr><td>r</td></tr>
<tr><td>L</td><td rowspan="3">$N_{\max} = \lfloor \sqrt{N} \rfloor$</td><td rowspan="3">$k = \frac{mS}{N}$</td><td rowspan="2">$\gamma = k\lambda$</td></tr>
<tr><td>N</td></tr>
<tr><td>S</td><td>$\gamma_{\max} = k\lambda_{\max}$</td></tr>
</table>

c. parameters that have to be chosen (tertiary parameters): arrival rates ratio: α, (thus we have average arrival rate: λ), average generation rate: γ, maximum generation rate: γ_{max}.

The table 3.3 gives the parameters that are used for time delivery calculation and which are put in the table using the above classification.

During the design processof a sensor network application, we have to approximate the values of the secondary parameters and to choose the values of the tertiary parameters. The values of the secondary parameters will be changed after the deployment of the unattended sensor networks. If the time entity is used with the mechanisms described below, the worst consequence of the deployment will be that some sensors will not participate with information for the monitored phenomenon. Figure 3.13 shows this case.

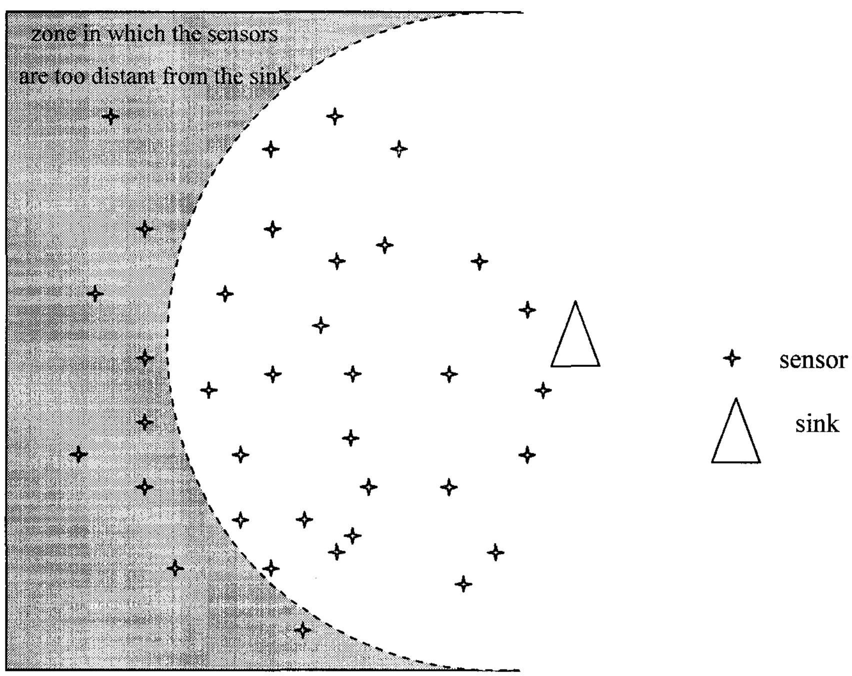


Figure 3.13 Sensors that will participate at the reported information

When the parameter α is chosen, the value of the imposed average arrival rate λ

is calculated based on the maximum arrival rate λ_{max}. But λ_{max} is approximate during the design process, and after the network deployment the value of λ_{max} has changed. This has as consequence a different value of the delay per hop. More over, the number of hops for data units to go through the network could be changed. Thus for some sensors, the delivery time (travel time) would be greater than the imposed deadline.

The sensors that are inside the circle (figure 3.13) will participate at the reported information It is obviously that in a real sensor deployement the border is not a perfect cicle. The delivery time of the data units for these sensors will be shorther than the imposed deadline. The sensors that are outside of the circle will send also their data units about the monitored phenomenon but their delivery time will exceed the deadline and therefore the time entity mechanisms will discard them.

3.4.2 Mechanism Overview

Taking into account the time delivery analysis and the principle of the time transport protocol presented in the first subchapter (3.1), three time mechanisms will be implemented in the time entity:

a. filtering

b. scheduling

c. traffic shaping

Filtering is a mechanism that will discard all the data units that do not meet the imposed deadline. When a data unit arrives at the time entity of every sensor, it is verified if it has exceeded the deadline. If this happens, the data unit is discarded. If not, the data unit will go to other time mechanisms.

Scheduling is the mechanism that changes the transmission order between the data units to help the data units with less budget of time to be sent before the data units with more budget of time. This mechanism will help the data units to meet their deadline.

Traffic shaping is a mechanism that ensures an average number of data units to be sent in a time unit. This mechanism ensures the imposed average data unit arrival λ of every sensor, in order to keep the delay data unit per hop at a specific value.

3.4.3 Filtering Mechanism

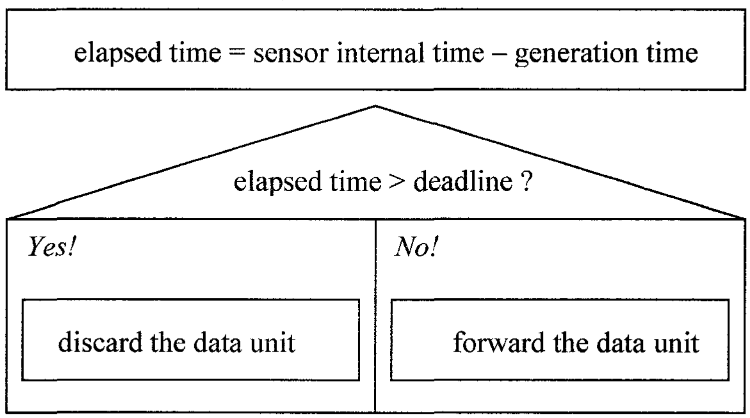


Figure 3.14 Algorithm of the filtering mechanism

Discarding the data units that have exceeded their deadline alleviates the network from the data units that are useless to be carried up to the end. The useless data unit will be discarded along the path, immediately after they miss the deadline. To do this, every sensor will be aware of deadline and the mechanism will compare the time elapsed from the data unit generation time up to the verification time. If the deadline is greater than the elapsed time, the data unit will not be discarded but will be forwarded to the next hop. If the deadline is shorter then the elapsed time, the data unit will be discarded (figure 3.14).

These require that every sensor should be time aware and the data units should transport the information about the generation time. All the sensors are time and localization aware. The data unit will have a field where the application layer will introduce the generation time. The time entity in every sensor will be aware of the deadline that has to be imposed at the data units. This information will be sent by the center base unit towards the sensors using the commands messages or could be loaded in the application layer before sensors deployment. In the figure 3.15 are represented the elements for filtering mechanism.

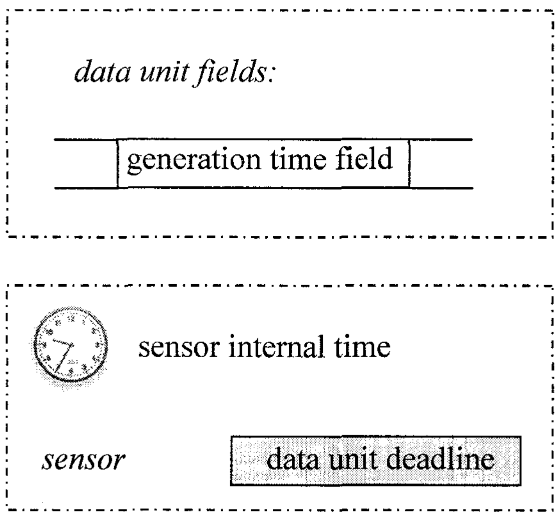


Figure 3.15 Required elements for filtering mechanism

3.4.4 Scheduling Mechanism

Scheduling is the mechanism that changes the transmission order to help data units to meet their deadline. To make this control the mechanism needs a function to evaluate every data unit and a system of queues. The function, named priority function, will calculate the priority for the data unit and the mechanism will put it in a queue corresponding to its priority. As there are data units in the queues system the mechanism will send first the data units from the queues which have higher priority and then data units from queues corresponding to lower priority. Thus, the time entity will help a data unit that is evaluated by the priority function as "late" to be sent toward the next hop before the data units that could wait. The priority function evaluates the data units locally thus from sensor to sensor along the path a data unit could change its priority.

Queue System

In the figure 3.16 is represented the time entity system of queues. The system includes 7 queues, numbered from 1 to 7. The queue number corresponds at the priority of the data units that will be put inside. The number 1 corresponds to the highest priority and the number 7 corresponds to the lowest priority. Thus, the data units in the queue number 1 will be send first, when this queue is empty, the mechanism will send the data units from the queue number 2 and so on up to the data units put in the queue number 7. If, during a data unit sending, another data unit appears in a higher priority queue, the time entity will first finish sending the data unit which was in the sending proceess. After this, the time entity will pass to send the data unit existing in the queue with higher priority. This policy is named “nonpreemptive” [21]. So, the time entity will verify only after every data unit sending if there are other data units in queues with higher priority. When the time entity starts to send a data unit from a queue all the queues with higher priority have to be empty.

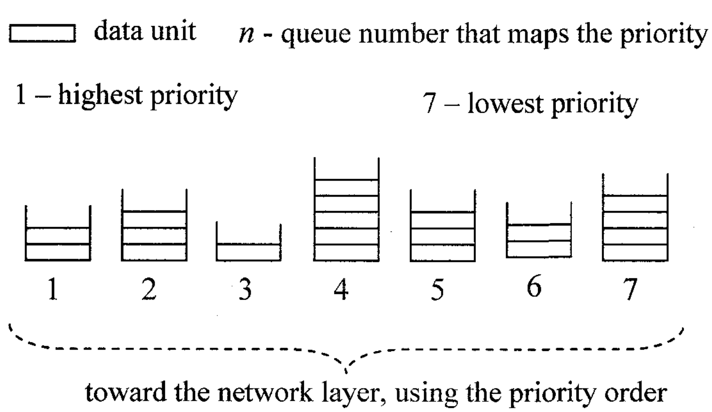


Figure 3.16 The time entity queue system

In this way, the time entity will help the data unit that are late to pass over the data units that have enough time budget.

Priority Function ζ (Zeta)

The priority function ζ has to evaluate every data unit and decides the priority number that it has to assign. The priority function will make the evaluation based on the time and distance. The time and distance evaluation are closely related, because the data unit has to be prioritized taking into account the elapsed time and the carried out distance. A data unit should neither finish the time budget, nor remain away from its destination. Therefore, it is necessary to buildup a function that evaluates the data unit from a spatio-temporal point of view.

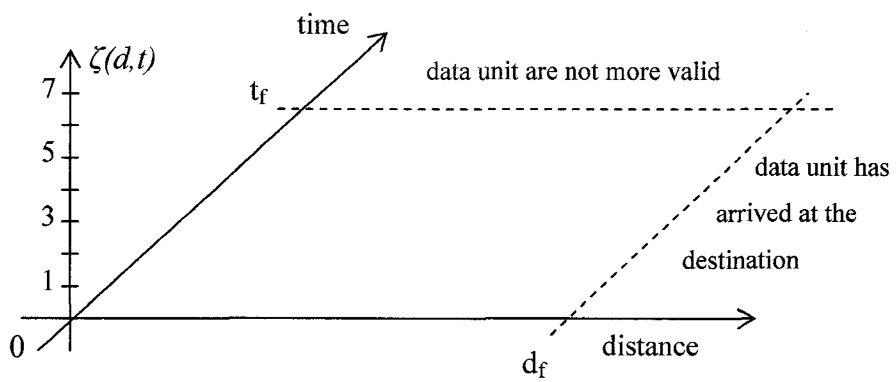


Figure 3.17 The priority function $\zeta(d,t)$

The priority function has two variables: distance and time $\zeta(d,t)$ (figure 3.17). When a data unit is traveling in the sensor network it will be (from time and distance point of view) in the rectangular zone delimited by the two axes, the t_f line and d_f line. If a data unit passes over the t_f line, the data unit will not be valid and it will be discarded. Therefore the t_f represents the data unit deadline. When the data unit passes the line d_f it has arrived at the destination, it has covered the entire distance between source and destination. As a consequence, the function $\zeta(d,t)$ is defined on $d \in (0,d_f)$ and $t \in (0,t_f)$, as in figure 3.17.

Every generated data unit starts the existence in the network at $d=0$ and $t=0$ and it has to cover d_f distance in maximum t_f time. The evaluation function will take 7 values corresponding to the 7 priorities. Based on the value of the priority function the time entity will decide in which queue (of the depicted queues system) the data unit will take place. Thus: $\zeta : [0, d_f] \times [0, t_f] \rightarrow \{1,2,3,4,5,6,7\}$

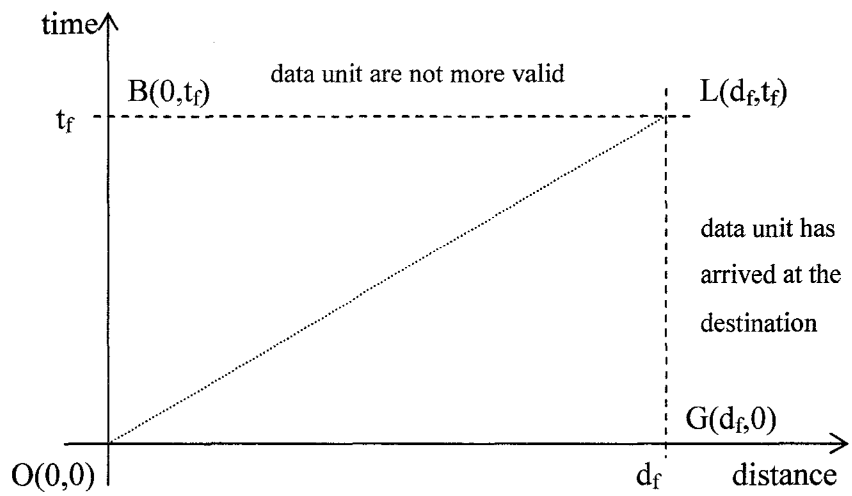


Figure 3.18 The evaluation function variable: distance and time

As it can be seen in the figure 3.18, when the data unit starts to travel in the network it will be at the point *O(0,0)*. Our aim is that the data unit will finish the network travel somewhere on the segment GL, because it has covered the distance that is between source and destination and the travel time is less than the deadline t_f. When the data unit arrives on the segment BL, the data unit will be discarded because it arrived somewhere in the network (not at the destination) but its time budget is finished. The point $G(d_f, 0)$ is the best point for the data unit because it arrived at the destination instantaneously when it was generated. The point $B(0,t_f)$ is the worst point that indicates that the data unit was not able to leave the source for a time equal with the deadline. The segment OL indicates that the data unit has a constant speed over the network and it is neither late nor early. The point L is the last point of arrival when a data unit covered the distance and is still valid. When the data unit is in zone OBL it is late and when it is in zone OLG it is early. Therefore, we propose the following divisions of the zone OBLG, as in figure 3.19:

Each zone is numbered from 1 to 7 and the number corresponds to the priority function value and the function value corresponds to the numbered queue in which the data unit will be put. If a data unit is evaluated that is in zone 3 (as an example), the priority function for this data unit will take the value 3 and it will be put in the queue number 3. A data unit that belongs to zone 1 will have the highest priority and will be sent before any other data unit that belongs to other zones. The data unit that belongs to zone 7 will have the smallest priority and will be sent after all other data units that belong to other zones. Every zone is delimited by four lines. For the zone delimiting, lines should be curve lines which model better the zone delimitations but they would be more difficult to be calculated. If the delimiting lines were elongated they would cross the axes in the points that are shown in the figure 3.19.

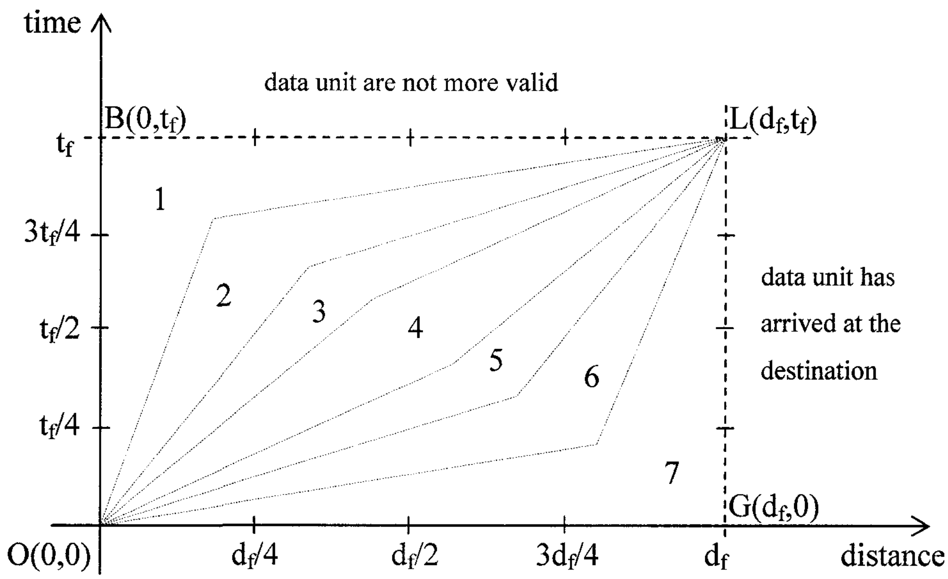


Figure 3.19 Evaluation zones

Let us note d_m and t_m the carried out distance and elapsed time from source to the sensor where the evaluation takes place, therefore:

$\zeta(d_m, t_m) = 1$ when:

$$t_m < t_f \text{ and } d_m > 0 \text{ and } t_m > \frac{1}{4}\frac{t_f}{d_f}d_m + \frac{3}{4}t_f \text{ and } t_m > 4\frac{t_f}{d_f}d_m;$$

$\zeta(d_m, t_m) = 2$ when:

$$t_m \le \frac{1}{4}\frac{t_f}{d_f}d_m + \frac{3}{4}t_f \text{ and } t_m \le 4\frac{t_f}{d_f}d_m \text{ and } t_m > \frac{1}{2}\frac{t_f}{d_f}d_m + \frac{1}{2}t_f \text{ and } t_m > 2\frac{t_f}{d_f}d_m;$$

$\zeta(d_m, t_m) = 3$ when:

$$t_m \le \frac{1}{2}\frac{t_f}{d_f}d_m + \frac{1}{2}t_f \text{ and } t_m \le 2\frac{t_f}{d_f}d_m \text{ and } t_m > \frac{3}{4}\frac{t_f}{d_f}d_m + \frac{1}{4}t_f \text{ and } t_m > \frac{4}{3}\frac{t_f}{d_f}d_m;$$

$\zeta(d_m, t_m) = 4$ when:

$$t_m \le \frac{3}{4}\frac{t_f}{d_f}d_m + \frac{1}{4}t_f \text{ and } t_m \le \frac{4}{3}\frac{t_f}{d_f}d_m \text{ and } t_m > \frac{4}{3}\frac{t_f}{d_f}d_m - \frac{1}{3}t_f \text{ and } t_m > \frac{3}{4}\frac{t_f}{d_f}d_m;$$

$\zeta(d_m, t_m) = 5$ when:

$$t_m \le \frac{4}{3}\frac{t_f}{d_f}d_m - \frac{1}{3}t_f \text{ and } t_m \le \frac{3}{4}\frac{t_f}{d_f}d_m \text{ and } t_m > 2\frac{t_f}{d_f}d_m - t_f \text{ and } t_m > \frac{1}{2}\frac{t_f}{d_f}d_m;$$

$\zeta(d_m, t_m) = 6$ when:

$$t_m \le 2\frac{t_f}{d_f}d_m - t_f \text{ and } t_m \le \frac{1}{2}\frac{t_f}{d_f}d_m \text{ and } t_m > 4\frac{t_f}{d_f}d_m - 3t_f \text{ and } t_m > \frac{1}{4}\frac{t_f}{d_f}d_m;$$

$\zeta(d_m, t_m) = 7$ when:

$$t_m \le 4\frac{t_f}{d_f}d_m - 3t_f \text{ and } t_m \le \frac{1}{4}\frac{t_f}{d_f}d_m \text{ and } t_m > 0 \text{ and } d_m < d_f.$$

As the evaluation function $\zeta(d,t)$ discards the data units that have the elapsed time t_m greater than the deadline: t_f, this mechanism has the same scope as the filtering mechanism. The scheduling mechanism covers the filtering mechanism. For this reason, only one mechanism, named scheduling-filtering mechanism will be implemented.

The number of queues in the queue system could be decreased, thus the priority function values number has to be decreased. In the Annex 1, we presented a possible priority function with four values. This number corresponds with the number of the queues in queue system.

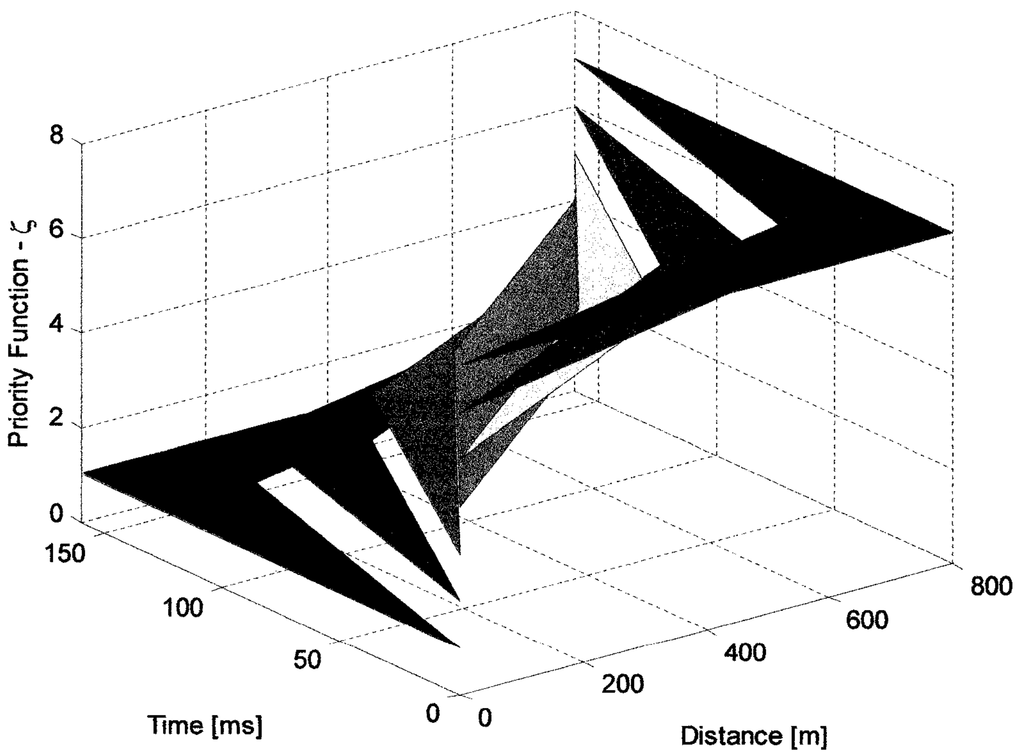


Figure 3.20 The priority function ζ, with seven values

Mechanism Elements

For this mechanism, every sensor should be time and position (location) aware, and have the data unit deadline and destination address. The data unit has the source address and the time generation fields. The sensor time, deadline and the time generation fields are necessary to evaluate the data unit in time as in the figure 3.15 for filtering mechanism. The sensor position, destination address and the source address are necessary to evaluate the data unit in space (figure 3.21). The last three elements are: the position of the source node, of the destination node and that of the transit sensorthat is doing the evaluation.

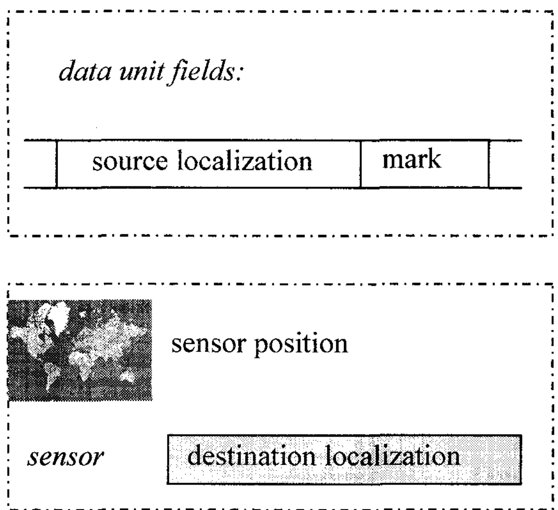


Figure 3.21 Required elements for spatial evaluation

The localization corresponds, in the sensor networks, at the global logical address for the network layer. Using the localizations, a sensor is able to calculate distances for the spatial evaluation of the data unit. The mechanism has to calculate the distance between source and destination, and between its position (the sensor position that makes the evaluation) and the destination.

In the sensor network there are also data units that do not carry information about the monitored phenomenon and thus they do not have a deadline. As a consequence these data units should not be evaluated spatio-temporal and should by-pass this mechanism. These data units will be put directly in the lowest priority queue (queue number 7). Therefore, the last element necessary for this mechanism is the type of the data unit (mark field), to distinguish between the data units that are critical or non-critical from the time point of view (figure 3.21).

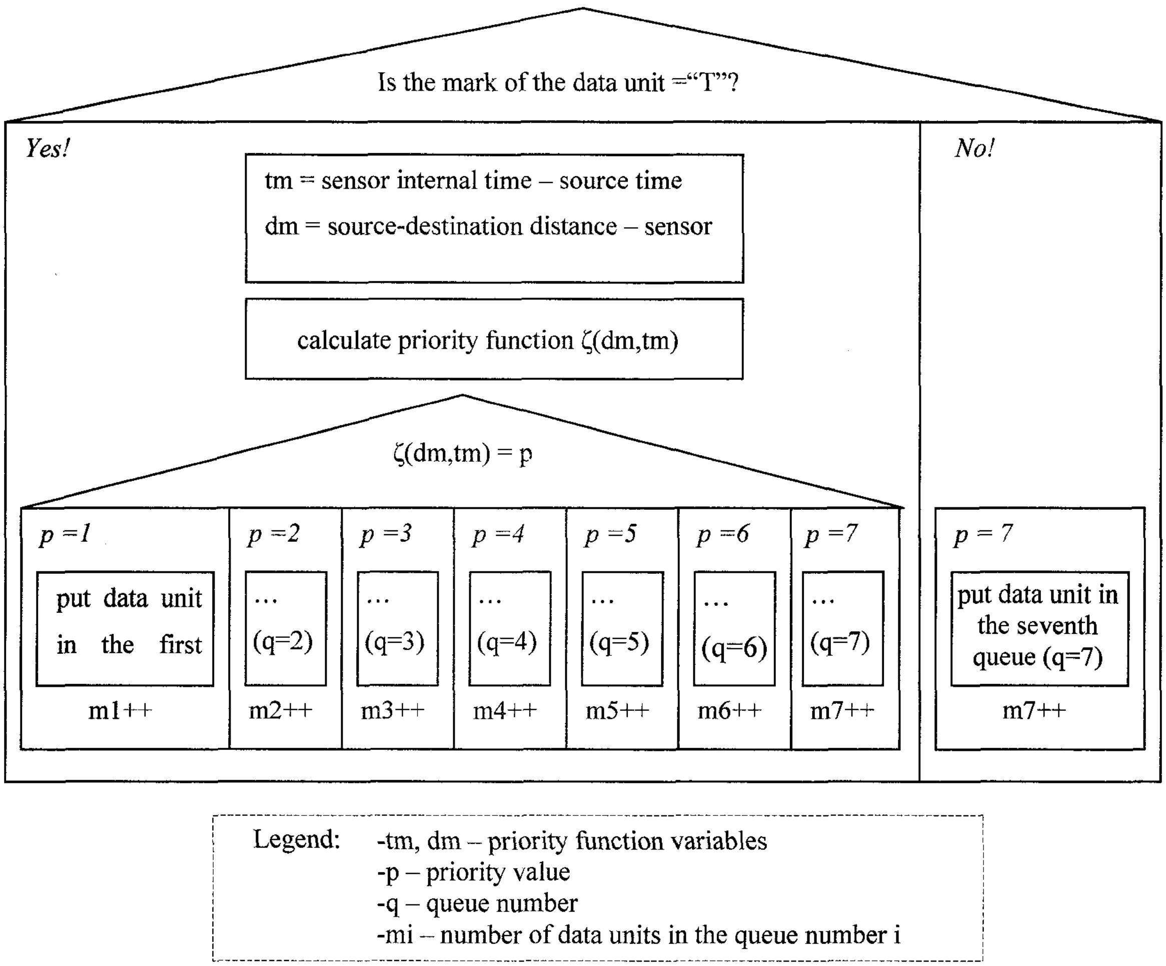


Figure 3.22 Queue selection part of the scheduling mechanism algorithm

Scheduling Mechanism Algorithm

The algorithm of the scheduling mechanism has two parts as is shown in the figures 3.22 and 3.23. In the first part, the data unit is evaluated with the priority function $\zeta(d_m, t_m)$, and based on the priority is put in the corresponding queue.

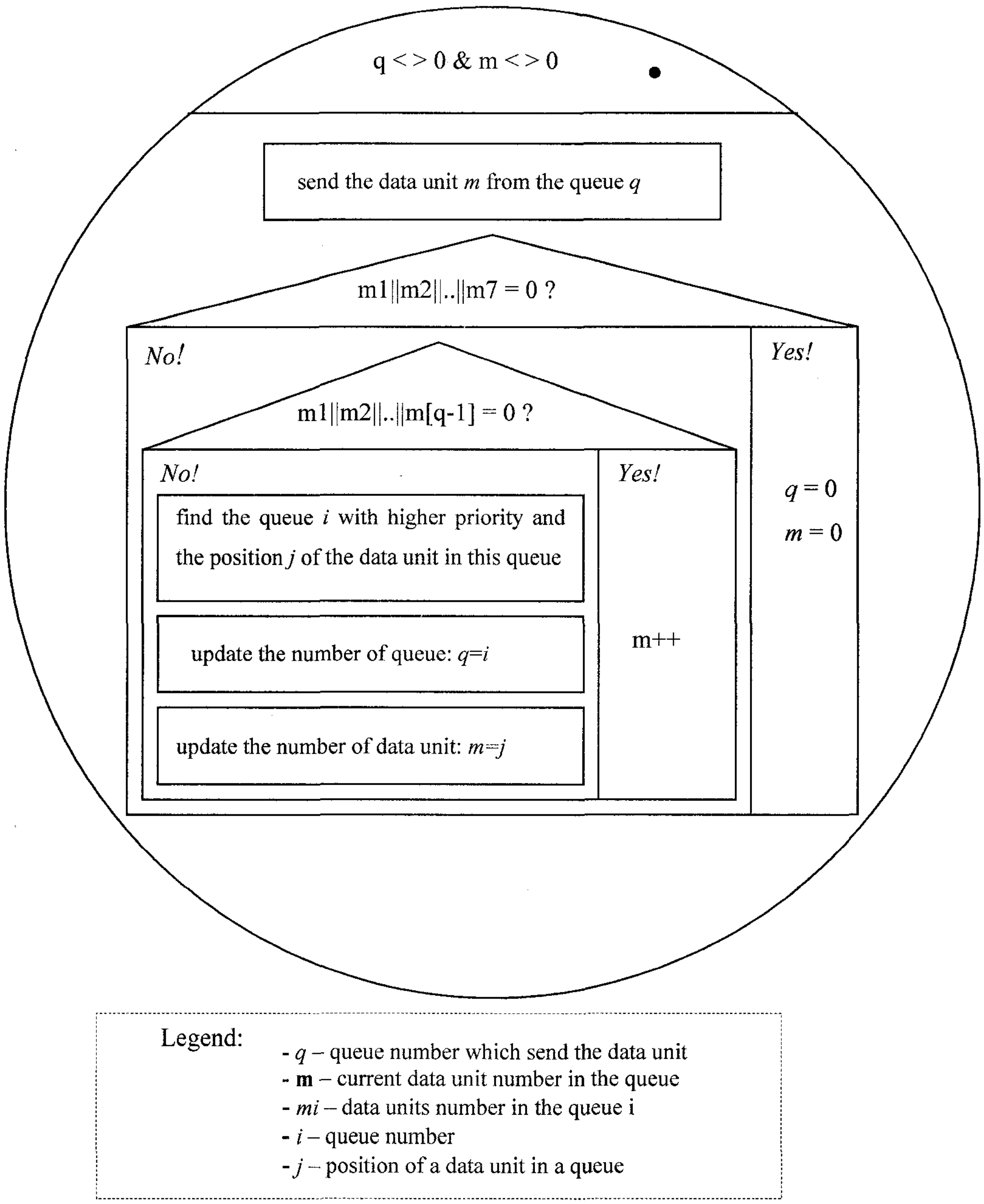


Figure 3.23 Data unit sending part of the scheduling mechanism algorithm

When a data unit passes this mechanism, it is verified the data unit type. If the data unit is marked with "T" (time treatment) it will be submitted at the time treatment (figure 3.22). If the data unit is not marked with "T" it will be put directly in the queue number 7 which has the lowest priority. If the data unit has a time treatment, the algorithm will calculate the elapsed time t_m from the generation of it and the covered distance d_m, based on the mechanism elements carried by the data unit and the sensor time and localization. Using the t_m, d_m, and the priority function $\zeta(d_m, t_m)$, the algorithm calculates the priority value for the data unit. Based on the priority value, the data unit is put in the corresponding queue. Eventually, the algorithm updates the number of data units m_i that exist in the queue in which the data unit has been put (figure 3.22).

In the second part of the algorithm, figure 3.23 represents the sending part, when the data units are sent toward the network layer in the order of their priority.

The algorithm remains in a loop until all the data units are sent out. This could be verified when the variable that keep the number of queue q and the number that keep the number of the data unit that has to be sent m are not zero. If the variables q and m are not zero, the algorithm sends the data unit indicated by variable q and m. After the data unit is sent, the algorithm verifies if there are any data units to be sent in all queues (by calculation the value of the equation $m_1 \,||\, m_2 \,||..||m_7$ where m_i is the number of the data units in the queue i. If there are more data units to be send (the queues are not empty), the algorithm verifies if there are other data units in queues with higher priority than the current queue. This verification is made by checking the values of the number of data units (given by m_1, m_2.. m_{q-1}). If one of these variables is not zero, the algorithm has found a queue with more priority that is not empty. In this case, the variables q and m are upgraded with the number of new queue and the new number of data unit in the queue that have to be sent at the next step. If all the queues with higher priority than the current queue are empty (the value of equation: $m_1 \,||\, m_2 \,||..||m_{q-1}$ is zero then only the variable m is upgraded and the algorithm will send the data unit indicated by variable q and m. If all the queues are empty, ($m_1 \,||\, m_2 \,||..||m_7 = 0$) the algorithm reset the variable p and m and the mechanism will stop data unit sending till a new data unit will be put in a queue (figure 3.23).

3.4.5 Traffic Shaping Mechanism

Traffic shaping mechanism ensures the average arrival rate λ of the data units for every sensor. If the sending of the data units per unit of time is controlled at the time entity, the same amount of the data units will be transmitted (per time unit) at the data-link layer which is the average arrival rate of the data units. Sending the data units by the transport protocol with a specific average rate, ensures the same average arrival at the data-link layer (figure 3.24). The position of this mechanism in the time entity is after the scheduling-filtering mechanism, just before the network layer as is shown in the figure.

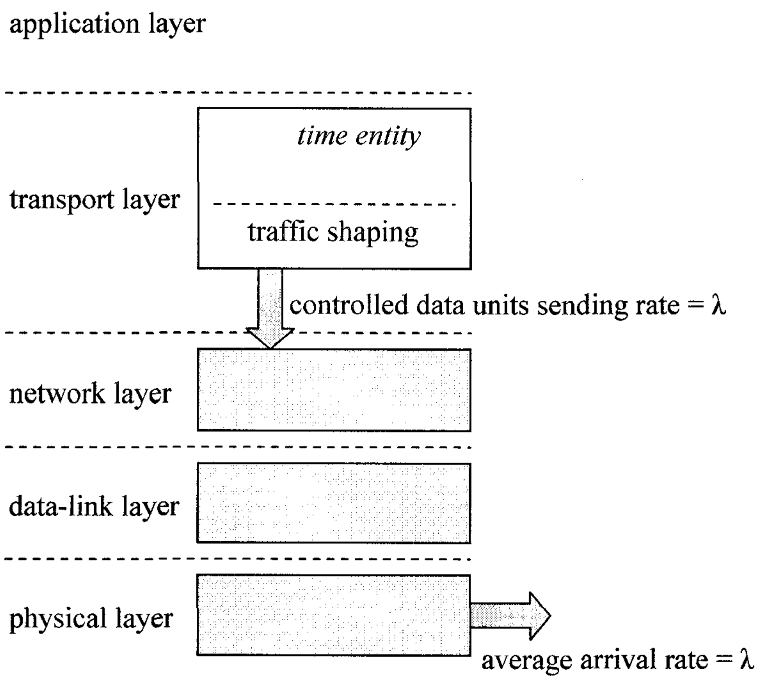


Figure 3.24 Traffic shaping mechanism will control the average arrival rate λ

The evaluation has shown that in order to have a bounded travel time over the network, it is necessary to control the average data unit arrival of every sensor. The value of the average data unit arrival is imposed for every sensor by commands sent by the central base unit. When too many data units are presented at the time entity, it is necessary to discard some, in order to keep the average unit arrival at the specific value. As in the time entity we have built a system of queues, it is possible to develop a discarding policy of the data unit. Thus, the remaining data units will provide necessary and sufficient information to derive the variation of the measured environmental phenomenon. For the simplicity of this protocol, we will chose to discard the first data unit that cannot enter in the queue system, indifferent of its priority or of its mark.

Token Bucket Algorithm

The traffic shaping mechanism is obtained using the token bucket algorithm as it is described in [20]. The token bucket algorithm is an improvement of the leaky bucket algorithm. The leaky bucket algorithm uses a queue and at specified interval time a data unit is sent out. When the queue is full, all the new arrived data units are discarded. This algorithm cannot ensure a good control when the data units have different length. Instead of this, the token bucket algorithm is used. In the figure 3.25 is shown the principle of the token bucket algorithm.

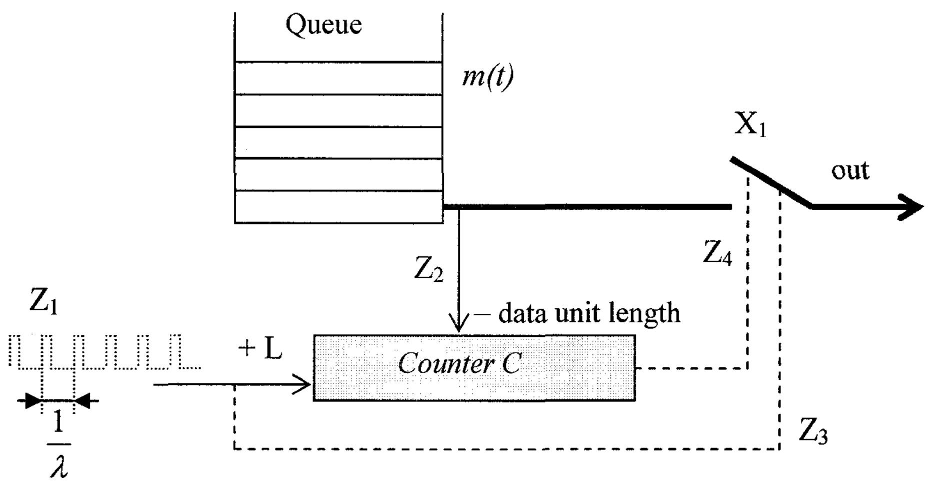


Figure 3.25 The principle of token bucket algorithm

The queue keeps the data units that wait to be sent. At every moment, the number of data units in the queue is *m(t)* which is function of time. The counter keeps the maximum number of bytes that can be sent out from the queue – let us note this number C. When a data unit has to be sent out, the length of the data unit (in bytes) is compared with the number of bytes indicated by the counter - C. If the data unit has a length greater than C, the data unit cannot be sent. If the data unit length is less than C, the data unit is sent out. If the counter has a number C that is greater than the added length of two or more data units, all these data units are sent out. When a data unit is sent out, the counter is decreased with the length of the sent data unit, as indicates the signal Z_2. The signal Z_1 is the signal that generates the clock ticks. These ticks are generated with the period of $\frac{1}{\lambda}$ and indicate the moments where a data unit has the possibility to be sent out. This is indicated in the figure by the controlling signal of Z_3 (Z_3 is the same signal as Z_1). Z3 controls the switch X_1. At the same ticks, the counter is incremented with L, where L is the average length of a data unit. When the counter arrived at zero the controlling signal Z_4 open the switch X_1 and the data units are impossible to be sent. The counter cannot be increased more than L_{max}, a value that indicates the maximum number of bytes that can be sent out at one tick.

Using this algorithm, we ensure the average data unit arrival rate at λ and an average arrival bytes rate at $L \cdot \lambda$ with maximum arrival bytes rate at $L_{\max} \cdot \lambda$. This algorithm is useful when the data units have different length values. If the successive data units have short lengths, they can be sent out in one tick. If the data unit has a longer length than the average length L, the data unit has to wait until the next tick, making an average of L bytes per ticks. If the queue is full and the average data unit arrival rate for the queue is greater than λ, some data units have to be discarded.

Another important policy for this mechanism is the decision to take when the queue is full (in our case the system of queues): what data unit should be discarded when the queue is full and there is another data unit that has to enter in the queue. Usually the last data unit is discarded (the one that wants to enter in the queue) but when the data units are differentiated by the priority we can chose another policy: to discard a data that is in the queue. For the simplicity for this protocol, we will choose to discard the last data unit, the one that has no room in the queue. This could be a subject for Bonaventura protocol improving.

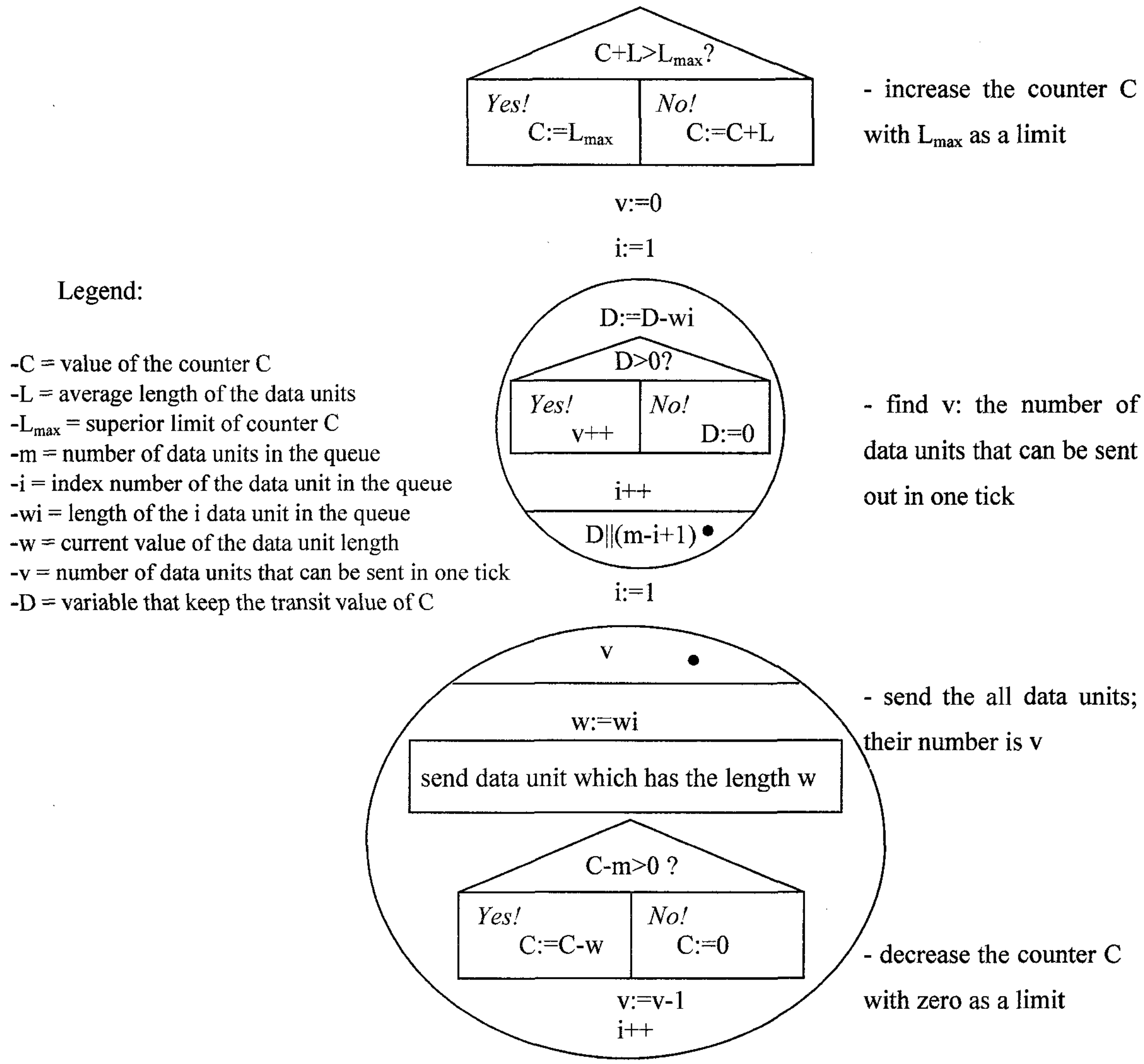


Figure 3.26 The algorithm of the traffic shaping mechanism at every tick

The required elements for this mechanism are: the value of λ – the average data unit arrival rate, the value of L - the average of the data unit length, and L_{max}, the maximum value of the number of bytes that has to be sent on a tick. Usually, the value of L_{max}, depends on L: as $L_{\max} = \beta \cdot L$, where β is a constant. Thus to implement this mechanism we need the value of λ and L. Theses values are sent by the center base unit toward the sensors using the command messages.

Mechanism Algorithm

In the figure 3.26 is shown the algorithm of the traffic shaping mechanism that is used at every tick. It has three parts: i) increase the counter C with the average length of the data units; ii) find the number of the data units that is possible to be sent in one tick; iii) send all the possible data units during the current tick. When the counter C is increased at every tick, its superior limit is L_{max}. When the queue is empty there is no data unit to be sent and at every tick, the counter is increased with L. In order to prevent to be sent too much data units on a tick, the counter is limited at L_{max}. When the number of data units v that can be sent during a tick is calculated, the variable D (which represents the transit value of C) that takes (only initially) the value of the counter C, is decremented down to zero. The loop to find v stops when D becomes zero or there are no more data units in the queue (m = 0). Knowing the number of the data units that can be sent during one tick interval, the data units are sent, and the counter C is decreased for every data unit with the value of the length of the data unit. The inferior limit of the C is zero. The loop for sending the data units stops when all data units are sent.

3.4.6. Time Entity Parts

The time entity of the transport protocol will include the two mechanisms: scheduling–filtering and traffic shaping. When traffic shaping mechanism will work with the scheduling–filtering mechanism, the queue of traffic shaping mechanism is represented by the system of queues of the scheduling–filtering mechanism. Therefore the two mechanisms will use the same queues system. The scheduling – filtering mechanism uses the queues system to put the evaluated data units in the corresponding queue and the traffic shaping will take the data units to send them toward the network layer in a controlled average arrival rate way as is represented in the figure 3.27.

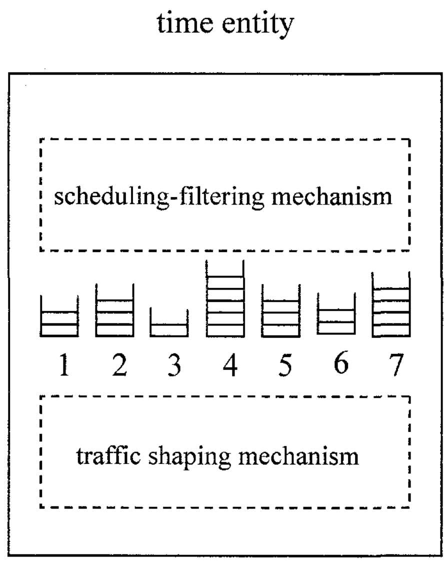


Figure 3.27 Constituent parts of the time entity

Chapter 4 Analytical Model

The aim for these analytical models is to build mathematical implementations for the two protocols: the Bonaventura protocol (BVP) – the designed transport protocol in this work – and for UDP protocol and compare their behavior in a real-time communication needed in the wireless sensor networks.

4.1 Introduction

In this subchapter we describe the comparison presumptions, the performance metrics and the notation used for the analytical model

4.1.1 Presumptions

For the consistency of the comparison of the analytical models, the two protocols have to be put under the same conditions and, therefore, the same sensor network will be considered and the same routing protocol will be used. Others presumptions are:

1. the sensors are fixed in the network field and are distributed randomly,

2. the communication system of the sensors is a very simple one, with a single communication channel. Thus, in a sensors neighboring (given by the radio range radius) only one single communication is possible; the other sensors should wait in order to seize the medium,

3. all the sensors generate data units periodically. The data units contain payload data for application; these data units have to be transmitted to the sink,

4. the data unit processing time at the layers 1, 2, and 3 is the same because the comparison imposes that the model should use the same protocols in the layers 1, 2 and 3.

4.1.2 Performance Metrics

The comparison of the two protocols behavior will be made for two performance metrics: travel time and throughput. Each data unit generated by a sensor network has a travel time and the entire sensor network has a throughput, depending on the used protocols stack.

The travel time starts in the moment when the data unit is delivered at the source sensor from the application layer to the transport protocol and ends when the transport protocol at the destination (sink) delivers the data unit to the application layer as is shown in the figure 2.1. This travel time will be calculated for the two protocols and the results will be compared.

The throughput is defined as the number of data units that arrive at the destination in a time unit. It is assumed that all these data units could be used by the application. In the real-time communication, not all data units that arrive at destination can be used. Therefore it is better to compare the data units that arrive in time: those that can be used by the application. This performance metric will be defined in the paragraph 4.3.5 "Timelyput Definition."

4.1.3 Notation

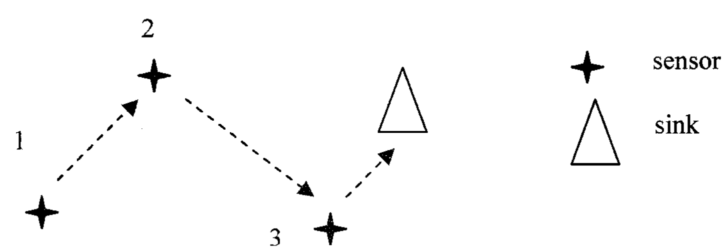


Figure 4.1 A path with three sensors for analytical model

In order to build the travel time model, we will take into consideration from a large sensor network a route that contain three sensors and a sink as in the figure 4.1 and we will try to find the travel time in the case of the two compared protocols.

The notations used in this chapter are:

- two indexes which indicate: *a)* if there is a letter (*h, p* or *r*) it means the delay data unit per hop (*h*), propagation time (*p*) and reception time (*r*) or if it is a number it means the layer at which the data unit is processed (*1, 2, 3*, or *4*) and *b)* the protocol: *U* stands for UDP and *B* stands for BVP

- one exponent which indicates the node where the time is considered and it is put in round parenthesis

4.2 Travel Time Analysis

We make the travel time analysis describing the travel time components for the two compared protocols.

4.2.1 UDP Travel Time

Transit sensor

The data unit is received by the communication system and it will pass and it will be processed up to the layer three. This time is the same for all the data units and is noted as $T_{1\text{-}3}$. At the third layer the data unit is analyzed in order to be routed; this time is noted as T_3. Then the data unit will be delivered to the second layer and then to the first layer. This time will be noted as $T_{3\text{-}1}$ and will be considered as the same for all data units. The order of the data units from their receiving in a sensor up to their sending towards the next sensors is kept. In the moment that the data unit has to be sent towards the next sensor the sensor has to seize the medium. Thus the data unit has to wait in a queue. This is the moment in which the UDP protocol – more precisely the network is unable to control the data unit travel time because the waiting time could be very long for a real-time communication unit. This time is noted T_h and comprise also the transmission time toward the next node.

Generating sensor

It was stated that all the sensors are also nodes that generate data units. The application layer will deliver data units at the UDP. From this moment we have to calculate the travel time. When a data unit arrives at the transport protocol UDP, UDP will process it in a time noted with T_{4U}. Then the data unit is delivered to the third layer and at this level it meets the data units that are in transit in the same node. It is processed to be routed which will take the time T_3. The rest of the model is the same as in transit sensor: data units will be delivered to the second and then to the first layer and transmitted toward the next sensor.

Sink node

At the end of the travel is the sink node which receives all the data units generated by the sensor network. The data unit is received and then delivered up to the third layer in a time noted by $T_{1\text{-}3}$. At the third layer, the data unit is analyzed in a time noted T_3 and it is delivered to the transport layer because the data unit reached its destination. Here the protocol UDP works and we will note the time to analyze the data unit and to transmit it to the application layer with T_{4U}.

UDP Travel Time Components

When the node three generates (figure 4.1) a data unit, the travel time is composed by the following times:

$T_{4U}^{(3)}$ = data unit processing time by the UDP at source sensor (3)

$T_3^{(3)}$ = data unit processing time by the third layer for routing

$T_{3-1}^{(3)}$ = data unit passing time from the third layer to the first layer in order to be transmitted

$T_{hU}^{(3)}$ = delay data unit to wait at the queue and to be transmitted (hop delay)

$T_{p}^{(3-2)}$ = propagation time between the node (3) and (2)

$T_{r}^{(2)}$ = reception time in node (2)

$T_{1-3}^{(2)}$ = data unit passing time from the first layer to the third layer

$T_{3}^{(2)}$ = data unit processing time at the third layer for routing

$T_{3-1}^{(2)}$ = data unit passing time from the third layer to the first layer in order to be transmitted

$T_{hU}^{(2)}$ = delay data unit to wait at the queue and to be transmitted (hop delay)

$T_{p}^{(2-1)}$ = propagation time between the node (2) and (1)

$T_{r}^{(1)}$ = reception time in node (1)

$T_{1-3}^{(1)}$ = data unit passing time from the first layer to the third layer

$T_{3}^{(1)}$ = data unit processing time by the third layer for routing

$T_{3-1}^{(1)}$ = data unit passing time from the third layer to the first layer in order to be transmitted

$T_{hU}^{(1)}$ = delay data unit to wait at the queue and to be transmitted (hop delay)

$T_{p}^{(1-s)}$ = propagation time between the node 1 and sink (s)

$T_{r}^{(s)}$ = reception time in sink node (s)

$T_{1-3}^{(s)}$ = data unit passing time from the first layer to the third layer

$T_{3}^{(s)}$ = data unit processing time by the third layer for routing

$T_{4U}^{(s)}$ = data unit processing time by the UDP at destination node: sink (s)

Therefore for a data unit generated by the node three, the travel time T_t will be the sum of all of described times:

$$T_{tU}^{(3)} = \sum_{a=1-4,h,p,r;c=1-3,s} T_{aU}^{(c)}$$

(Eq.4.1)

When node two generates a data unit, the travel time will be obtained in the same way as in the case of the third node. From equation (4.1) it is possible to obtain the travel time for a data unit generated by the second node:

$$T_{tU}^{(2)} = \sum_{a=1-4,h,p,r;c=1-2,s} T_{aU}^{(c)}$$

(Eq.4.2)

When the first node generates a data unit, the travel time is given by the following equation:

$$T_{tU}^{(1)} = \sum_{a=1-4,h,p,r;c=1,s} T_{aU}^{(c)}$$

(Eq.4.3)

4.2.2 BVP Travel Time

The UDP protocol runs only at the ends of the communication: at source node and at the destination node. In the transit nodes the data unit is handled only up to layer 3 in order to be routed. The BVP protocol runs in all the nodes during the travel, so a data unit will be processed in every node along the way. The data unit will arrive at the forth layer and will be processed.

Transit sensor

In a transit node, the data unit is received by the communication system and it passes from layer to layer up to the third layer. This passing time is the same for all data units and will be noted: $T_{1\text{-}3}$. The data unit will not be analyzed at the network layer but it will be delivered direct to the forth layer where the BVP protocol runs. We will consider that the passing of the data unit from the third layer to the fourth layer is very quick. As a consequence of the analysis made for the BVP protocol, the data unit will be put in one of queues implemented in the protocol and will be transmitted toward the third layer using the traffic shaping protocol. For this layer we will consider that the time spent by the data unit at this layer is composed by:

- T_{41} = analysis time: the data unit is analyzed by the BVP for spatial and temporal point of view and is introduced in a queue. The estimation of this time is made by the estimation of the number of instruction and the computation speed of the CPU sensor.

- T_{42} = waiting time in queue: in the case of a data unit having the highest priority, this time is given by the number of data units that exist in this queue. If a data unit has a lower priority, the waiting time is given by the number of data units that exist in all the queues with higher priority plus the number of data units that exist in its queue before it.

- T_{43} = emission time towards the third layer because of the shaping mechanism. The shaping mechanism ensures a specified number of data units to be emitted in the time unit.

As a consequence, at transport layer, in a transit node, the time to process a data unit is given by:

$$T_4 = T_{41} + T_{42} + T_{43} \qquad \text{(Eq.4.4)}$$

Then the data unit will be delivered at the third layer where it will be analyzed for routing. The data unit will pass from the third layer to the first layer in the time noted $T_{3\text{-}1}$. The data unit will wait in order to be transmitted. This time is noted by T_{hB}, it also comprises the transmission time and its estimation is determined by the number of sensors in the radio range which depends on the average sensor density in the field.

Generating sensor

In the case of the generating sensor, the time to process the data unit by the BVP protocol is limited to time T_{42} and T_{43} because the time to analyze a data unit does not exist: the data unit is put directly in queue with the last priority. Thus at the generating node the processing time at the transport layer is:

$$T_4 = T_{42} + T_{43} \qquad \text{(Eq.4.5)}$$

The rest of the time components remain as in the transit node.

Sink node

In the case of sink node, the processing time of the data unit by the BVP protocol is composed only by the first component: T_{41} because the data unit will be delivered directly to the application level after spatial and temporal analysis and therefore the components T_{42} (waiting in queue) and T_{43} (emission time) do not exist. Thus at the sink node the processing time at the transport layer is:

$$T_4 = T_{41} \qquad \text{(Eq.4.6)}$$

BVP Travel Time Components

The considered paths remain the same as were for UDP analysis (figure 4.1). When the node three generates a data unit, its the travel time is composed by the following times:

$T_{4B}^{(3)}$ = data unit processing time by the BVP at source sensor (3)

$T_{3}^{(3)}$ = data unit processing time by the third layer for routing

$T_{3-1}^{(3)}$ = data unit passing time from the third layer to the first layer in order to be transmitted

$T_{hB}^{(3)}$ = delay data unit to wait at the queue and to be transmitted (hop delay)

$T_{p}^{(3-2)}$ = propagation time between the node (3) and (2)

$T_{r}^{(2)}$ = reception time in node (2)

$T_{1-3}^{(2)}$ = data unit passing time from the first layer to the third layer

$T_{4B}^{(2)}$ = data unit processing time by the BVP at transit sensor (2)

$T_{3}^{(2)}$ = data unit processing time by the third layer for routing

$T_{3-1}^{(2)}$ = data unit passing time from the third layer to the first layer in order to be transmitted

$T_{hB}^{(2)}$ = delay data unit to wait at the queue and to be transmitted (hop delay)

$T_{p}^{(2-1)}$ = propagation time between the node (2) and (1)

$T_{r}^{(1)}$ = reception time in node (1)

$T_{1-3}^{(1)}$ = data unit passing time from the first layer to the third layer

$T_{4B}^{(1)}$ = data unit processing time by the BVP at transition sensor (1)

$T_{3}^{(1)}$ = data unit processing time by the third layer for routing

$T_{3-1}^{(1)}$ = data unit passing time from the third layer to the first layer in order to be transmitted

$T_{hB}^{(1)}$ = delay data unit to wait at the queue and to be transmitted (hop delay)

$T_{p}^{(1-s)}$ = propagation time between the node (1) and sink (s)

$T_r^{(s)}$ = reception time at sink node (s)

$T_{1-3}^{(s)}$ = data unit passing time from the first layer to the third layer

$T_3^{(s)}$ = data unit processing time by the third layer for routing

$T_{4B}^{(s)}$ = data unit processing time by the BVP at destination node (s)

The travel time for a data unit generated by the node (3) will be the sum of all of described times:

$$T_{tB}^{(3)} = \sum_{a=1-4,h,p,r;c=1-3,s} T_{aB}^{(c)} \quad \text{(Eq.4.7)}$$

It is possible to write the travel time generated by nodes (2) or (1) in the same way as for UDP:

$$T_{tB}^{(2)} = \sum_{a=1-4,h,p,r;c=1-2,s} T_{aB}^{(c)} \quad \text{(Eq.4.8)}$$

and respectively:

$$T_{tB}^{(1)} = \sum_{a=1-4,h,p,r;c=1,s} T_{aB}^{(c)} \quad \text{(Eq.4.9)}$$

4.2.3 Travel Time Comparison

If we take into the consideration the presumptions made in the paragraph 4.1.1, we can consider that the following times in the travel times are equal and are noted the same:

$T_{3U}^{(1)} = T_{3B}^{(1)} = T_3^{(1)}$ = data unit processing time at the third layer in node (1)

$T_{3U}^{(2)} = T_{3B}^{(2)} = T_3^{(2)}$ = data unit processing time at the third layer in node (2)

$T_{3U}^{(3)} = T_{3B}^{(3)} = T_3^{(3)}$ = data unit processing time at the third layer in node (3)

$T^{(1)}_{3-1U} = T^{(1)}_{3-1B} = T^{(1)}_{3-1}$ = data unit passing time from the third layer to the first layer in order to be transmitted from node (1)

$T^{(2)}_{3-1U} = T^{(2)}_{3-1B} = T^{(2)}_{3-1}$ = data unit passing time from the third layer to the first layer in order to be transmitted from node (2)

$T^{(3)}_{3-1U} = T^{(3)}_{3-1B} = T^{(3)}_{3-1}$ = data unit passing time from the third layer to the first layer in order to be transmitted from node (3)

$T^{(2)}_{rU} = T^{(2)}_{rB} = T^{(2)}_{r}$ = reception time in node (2)

$T^{(1)}_{rU} = T^{(1)}_{rB} = T^{(1)}_{r}$ = reception time in node (1)

$T^{(s)}_{rU} = T^{(s)}_{rB} = T^{(s)}_{s}$ = reception time at sink (s)

$T^{(3-2)}_{pU} = T^{(3-2)}_{pB} = T^{(3-2)}_{p}$ = propagation time between nodes (3) and (2)

$T^{(2-1)}_{pU} = T^{(2-1)}_{pB} = T^{(2-1)}_{p}$ = propagation time between nodes (2) and (1)

$T^{(1-s)}_{pU} = T^{(1-s)}_{pB} = T^{(1-s)}_{p}$ = propagation time between nodes (1) and sink (s)

$T^{(2)}_{1-3U} = T^{(2)}_{1-3B} = T^{(2)}_{1-3}$ = data unit passing time from the first layer to the third layer in node (2)

$T^{(1)}_{1-3U} = T^{(1)}_{1-3B} = T^{(1)}_{1-3}$ = data unit passing time from the first layer to the third layer in node (1)

$T^{(s)}_{1-3U} = T^{(s)}_{1-3B} = T^{(s)}_{1-3}$ = data unit passing time from the first layer to the third layer in sink (s)

Equivalent Model

For the reason that the times in previous paragraph are equals, we will eliminate these components from the travel time estimation because they do not give any new information for this model. Therefore we will consider as part of the model only the times that are different between the two protocols.

For the UDP protocol, a data unit generated in the third node will have the following the travel time (the equivalent of the 4.1 equation):

$$T_{tU}^{(3)} = T_{4U}^{(3)} + T_{hU}^{(3)} + T_{hU}^{(2)} + T_{hU}^{(1)} + T_{4U}^{(s)} \qquad \text{(Eq.4.10)}$$

For the BVT protocol, a data unit generated in the third node will have the following travel time is (the equivalent of the 4.7 equation):

$$T_{tB}^{(3)} = T_{4B}^{(3)} + T_{hB}^{(3)} + T_{4B}^{(2)} + T_{hB}^{(2)} + T_{4B}^{(1)} + T_{hB}^{(1)} + T_{4B}^{(s)} \qquad \text{(Eq.4.11)}$$

It can be seen that in the case of the UDP protocol, the travel time estimation is composed by two processing times at source and at destination: $T_{4U}^{(3)}$ and $T_{4U}^{(s)}$ plus the delays data unit per hop: $T_{hU}^{(c)}$. It is possible to generalize the equation (4.10) and we can write for a node j with a path with "C" hops:

$$T_{tU}^{(j)} = T_{4U}^{(C)} + T_{4U}^{(s)} + \sum_{c=1}^{C} T_{hU}^{(c)} \qquad \text{(Eq.4.12)}$$

In the case of protocol BVP, the travel time estimation is composed of:

- the processing time at layer 4 at source and at destination $T_{4B}^{(3)}$ and $T_{4B}^{(s)}$
- the processing time at layer 4 for all transit nodes $T_{4B}^{(2)}$, $T_{4B}^{(1)}$
- the delays data unit per hop: $T_{hB}^{(c)}$.

It is possible to generalize the equation (4.11) and we can write for a path with "C" hops that the travel time is:

$$T_{tB}^{(j)} = T_{4B}^{(C)} + T_{4B}^{(s)} + \sum_{c=1}^{C-1} T_{4B}^{(c)} + \sum_{c=1}^{C} T_{hB}^{(c)} \qquad \text{(Eq.4.13)}$$

If the equations 4.5 and 4.6 are used it is possible to write the sum of the times spent by a data unit at layer 4 at generating node (j) and at the sink node (s) as:

$$T_{4B}^{(C)} + T_{4B}^{(s)} = T_{42B}^{(C)} + T_{43B}^{(C)} + T_{41B}^{(s)}$$

The generating node *(j)* is the same with the "*C*" node , because "c" is the first node in the path.

It is possible to make the following approximation: $T_{41B}^{(s)} = T_{41B}^{(C)}$ which means that the evaluation process time at transport layer at source *(C)* and at the destination *(s)* in the case of protocol BVP is the same. Thus, the equation 4.13 becomes:

$$T_{tB}^{(j)} = \sum_{c=1}^{C} T_{4B}^{(c)} + \sum_{c=1}^{C} T_{hB}^{(c)} \qquad \text{(Eq.4.14)}$$

As can be seen the two equations 4.13 and 4.14 comprise two kinds of times: transport protocol processing time and delays data unit per hop.

4.3 Throughput Analysis

In this subchapter we show that if we take the throughput as performance metric, the UDP protocol behaves better than BVP. We also show that the throughput is not a very good performance metric for a real-time communication. As a consequence, we defined a new performance metric: the timlyput which is a goodput for data units with time constraint.

4.3.1 Maximum Throughput Considerations

As it was stated in the second presumption, the communication system of the sensors is a very simple one with a single communication channel. We assume also, that the sink communication system (toward the sensor network) is also of the same kind. As the sensors in the network generate data units, these data units could be used only if they pass through the sink node. Regardless the quantity of the generated data units the throughput of the sensor network is given by the capacity of the sink to pass data unit toward the actors or toward the central base unit. As a consequence, the maximum throughput of a sink is given by $\theta_{\max} = \frac{W}{L}$; where W is the transmission rate in kbyte/s and L the length of the data unit in bytes. The measurement unit for the throughput is data units per second. This equation does not take into consideration the fact that the channel utilization is less that 100% due to the waiting time for medium seizing.

In a sensor network, a sensor data unit generation rate could be greater than the capacity of the communication system and as the number of sensors varies between some hundreds to some thousands, it is obvious that the sink nodes represent bottlenecks that limit the data units passing.

In order to increase this throughput it is possible to use more than one sinks (S the number of sinks) and therefore the maximum throughput in the network should become: $\theta_{\max} = \frac{SW}{L}$. This equation has another limitation than the channel utilization as it was shown in the previous equation; the sinks have to be enough far away so that any two sinks could simultaneously receive data units. If two sinks are close, they could not receive simultaneously data units as the sensor nodes could not simultaneously seize the medium, thus the throughput is less than given in the previous equation. Another possibility to increase the sensor network throughput is the utilization of actors that wander through sensor field and receive directly data units from the sensors. This is better than having fixed sink nodes, but the analysis is beyond the scope of this work.

These considerations were made regardless of the protocol stack used for network communication. Different throughputs will be obtained when using different protocol stacks it but these values cannot overpass the maximum throughput given into the above equations.

The consequence of the throughput limitation at the sink is this: it is useless for sensors to have high data unit generation rate because this will cause only the congestion of the network. In the paragraph 3.3.4 "The Average Generation Rate" was obtained the maximum generation rate for a sensor taking into account the maximum throughput of the network (more precisely the maximum sinks throughput). The maximum generation rate for a sensor is $\gamma_{\max} = \frac{SW}{NL}$ (Eq.3.18) where N represents the number of sensors in the network.

In order to analyze the influence of the transport protocol over the throughput we will take into consideration two cases:

a). the sensors have an average generation rate of data units more than the maximum value given in the equation 3.18: $\gamma \geq \frac{SW}{NL}$

b). the sensors have an average generation less than the value given in the equation 3.18. $\gamma \leq \frac{SW}{NL}$

4.3.2 UDP Throughput Analysis

In the case of a network where every nodes generates more than the maximum generation rate, the communication system will be exceeded and all the nodes will work under congestion. The model is given in the paragraph 3.2.2.1. Every node will try to send its data units to next node, established by routing protocol, but the capacity of the receiving node is exceeded. The UDP protocol cannot control this situation and the networks protocols even if they have had congestion control, the congestion would not be alleviated because of the high quantity of the data units that are generated. The consequences of this situation are:

- all nodes are in congestions

- large delays per hop are generated by the waiting time in transmission queue
- transmission buffers are always full (second layer)
- destroyed data units
- maximum throughput is limited by the sink (or sinks)

In the case of a network where every node generates less than the maximum generation rate, the network could transport all the data units. Every node will try to send its data units to the next node and the delay will be small. But the field deployment of the sensors is randomly and in zone with high sensor density the congestion phenomenon could appear. As UDP has no control over the network congestion, only the network protocols could handle this situation, there is a probability to generalize the congestion to the entire network.

In this situation it will be:

- short delays in zone without congestion
- possibility of congestions which imply longer delays
- maximum throughput limited by the network generation rate of the data units.

4.3.3 BVP Throughput Analysis

a). In the case of a network where every node generates more than the maximum generation rate, the BVP mechanisms will limit the number of data units that will be let towards the network layer. The traffic shaping mechanism eliminates all the data units generated over the limit imposed by this mechanism. Usually the limit of this mechanism is less than the maximum generation rate. The scheduling-filtering mechanism will let the data units that have a better priority (that came from the previous nodes). As the sensor deployment is randomly on the field there are possible congestions and thus large delays. If data units will have large delays in some zones of the network, the scheduling-filtering mechanisms will prioritize them and send them before other data units. If the deadline is exceeded by these data units, the scheduling-filtering mechanism will discharge them and thus alleviate the network. The consequences of this situation are:

- possible congestions given by the sensor deployment
- data unit delays per hops given by a network without congestion
- queue buffers at BVP level (layer 4) always full but transmission buffers not full (second layer)
- destroyed data units: that exceed the traffic-shaping mechanism limit
- maximum throughput is limited by BVP traffic shaping mechanism

b). In the case of a network where every node generates less than the maximum generation rate, BVP protocol discards the data unit that will have a travel time grater than the deadline. This phenomenon could appear as a consequence of possible congestions.

The consequences of this situation are:

- data unit delays given by a network without congestion
- possible congestions given by the sensor deployment
- maximum throughput is less than the network generation rate or by the BVP traffic shaping mechanism (which is smaller) because of possible discarded data units

4.3.4 UDP and BVP Throughput Comparison

If throughputs are compared, we can easily see that UDP behaves better than BVP protocol. When the sensors have an average generation rate of data units more than the maximum generation rate value (the equation 3.18), the throughput is:

- limited by the sink in case of UDP
- limited by the traffic shaping mechanism of BVP

When the sensors have an average generation less than the maximum value, the throughput is:

- limited by the sensor generation rate of the data units in the case of UDP
- limited by the sensor generation rate or by the BVP traffic shaping mechanism

In both cases the UDP throughput is greater than the BVP throughput.

4.3.5 Timelyput Definition

This comparison has taken into account only the throughput as the number of data units per time unit that are passed by sinks nodes toward the central base unit. In a real-time communication, the deadline determines which data units could be used or not by the application. As a consequence, in a real-time communication it will be more important to compare the throughput of the data units that could be used by the application, that is, the throughput of the data units that arrived at the destination before the deadline.

In the specialized literature the term "goodput" is defined as the throughput at the application level i.e. the number of useful data units per time unit forwarded by the network, excluding the protocols overhead.

In the same way, we can create and define the term "timelyput", θ_D, as the number of data units per time unit in a real-time communication that arrive at the application layer destination in time – the time travel over the network is less than the deadline (D):

$$\theta_D = \frac{n_{T \le D}}{\Delta t} \qquad \text{(Eq.4.15)}$$

In this equation, the $n_{T \le D}$ is the number of data units whose travel time over the network is less than a deadline value and Δt is the considered period of time.

The timelyput performance metric of the two compared protocols UDP and BVP is more useful in a real-time communication than the throughput metric.

4.3.6 UDP and BVT Timelyput Comparison

When the sensors have an average generation rate of data units more than the maximum generation rate value (equation 3.18) the timelyput could be:

- zero in case of UDP, because all the data unit will have long travel times as all nodes work in severe congestion. The travel time will exceed the deadline and the number of data units that will arrive in time per time unit will be very small)

- limited by the traffic shaping mechanism of BVP and by the value of the deadline

As the deadline is smaller, the number of discarded data unit increases and the timelyput decreases. On the other hand, as it is shown in the paragraph 3.4.1 Time Equation Parameters" there are some sensors which could be too far from the sink and the data units from these sensors will have the travel time longer than the deadline. This situation will decrease also the timelyput. As the consequence the deadline value changes the timelyput value.

When the sensors have an average generation less than the maximum value, the timelyput is:

- limited by the network generation rate of the data units and by the value of deadline, in the case of UDP

- limited by the network generation rate or by the traffic shaping mechanism (which is smaller) and by the value of deadline

In this second case, the difference between two protocols is: in the case of large networks or in case of possible congestions, in the network will appear data units with a travel time already greater than the deadline. If the network uses UDP, these data units, useless in a real-time communication, will be carried out by the network up to the sink and application layer. The side effect of these useless data units is that they will delay the data units that are in time. Some data units, which normally could arrive in time, will be delayed up to the point that they also exceed the deadline and will become useless. In some conditions this situation could have a "snow ball" effect and compromise all the data units in the network and therefore the timelyput could be zero.

In the case of BVP, if data units with a travel time greater than the deadline appear, these data units will be eliminated by the BVP mechanisms. This situation will alleviate the network, the delays per hop will remain minimum and the timelyput will remain closer to the network generation rate or traffic shaping mechanism (which of two is smaller).

As a conclusion, even if the generation rate of the data units in the network is lesser than the sink nodes can carry, the random position of the sensors in the field could generate congestion. In this case, BVP will have a better timelyput.

4.4 Mathematical Model

As the two protocols are compared by the means of a mathematical model, and not by means of an entire network simulation some approximations are necessary.

4.4.1 Timelyput Approximation

First of all, we assume that all the data units generated by sensors could be carried by sink nodes. As the sink nodes do not became a bottleneck, the number of generated data units will be equal with carried data units by sink and thus no discarded data units. In this situation the throughput will be the same for both protocols but we have to show that the timelyputs are different.

The mathematical model is based on the generation of a large number of the data units and in our model the generation of the data units is not time dependent or between the numbers of data units that arrive in time. The timelyputs (which are time dependent) will be compared by means of: a) generation of the same amount of data units for both protocols and b) the percentage between the number of data units that arrive in time and the entire number of generated data units.

If n is the number of the generated data units (equal to the number of arrived data units), and Δt is the considered period of time, the throughput is:

$$\theta = \frac{n}{\Delta t} \qquad \text{(Eq.4.16)}$$

It is possible to define the percentage of the data units that arrive in time as:

$$\eta_D = \frac{n_{T \le D}}{n} \cdot 100 \qquad \text{(Eq.4.17)}$$

Using the equation 4.15, 4.16 and 4.17 we deduce that the timelyput is:

$$\theta_D = \theta \cdot \eta_D$$

As the throughput θ is the same for the two protocols, the comparison between the timelyput values for each protocol became the comparison in our model, between the percentages of the timelyput number of data units (or between the numbers of the data units that arrived in time, if the numbers of generated data units "*n*" is the same :

$$(\theta_{D,U} \leftrightarrow \theta_{D,B}) \equiv (\eta_{D,U} \leftrightarrow \eta_{D,B}) \equiv (n_{T \le D,U} \leftrightarrow n_{T \le D,B}) \qquad \text{(Eq.4.18)}$$

4.4.2 Travel Time Components

In the paragraph 4.2.3 "Travel time comparison" from the equations 4.12 and 4.14, it was shown that the travel time has two major components: transport protocol processing times and data unit delays per hop.

Transport Protocol Processing Time

The processing time terms in equation 4.12 and 4.14 are notated with $T_{4U}^{(C)} + T_{4U}^{(s)}$ for UDP and $\sum_{c=1}^{C} T_{4B}^{(c)}$ for BVT protocol. Transport protocol processing times are different for the two protocols. This difference comes from the following factors:

- the number of instructions necessary for the two protocols are different. As a rough estimation, the number of instructions of the BVP protocol is three times more than the UDP.

- BVP protocol utilizes a number of prioritized queues which can be seen as a small data base. The data base operation consists in putting an element (data unit) into the data base (queues), in finding an element with the maximum priority and in sending it toward the next layer. For these operations, there were developed different algorithms. These algorithms have different complexities but the majority of the algorithms has the complexity of order *O(n);* where *n* is the number of elements in the data base. The analysis of the queues number, and the length of the queues is not the scope of this work.

- BVP runs in every node along the path

All the factors enumerated indicate that the average time for transport protocol processing is greater for the BVP than the UDP. This time could be optimized in order that the difference between the two protocols could be smaller. Furthermore, as CPU becomes richer in resources, this difference is very little. Even if the difference between the two transport protocols is very small, it remains that that BVP has a processing time greater than UDP. This is the cost to have a real-time communication.

We consider that the time processing at the transport layer is at lest one order smaller than the transmission time and therefore we do not take this time into account for our mathematical model.

Data Unit Delays per Hop

These terms are notated with $\sum_{c=1}^{C} T_{hU}^{(c)}$ for UDP in equation 4.12 and with $\sum_{c=1}^{C} T_{hB}^{(c)}$ for BVP in the equation 4.14 ($c=1 \ldots C$). The data unit delays per hop are the delays of a data unit in order to be transmitted from a node to another. This time in queuing theory is named system time and comprises the waiting time and the transmission time as in [18]. In the paragraph 3.2.2 "Sensor Network Delay Analysis" it was shown the delay model and the data unit delay. In the equation the delay per hop is:

$$T_h = T_0 \frac{1}{1-\alpha}$$

where $T_0 = \frac{L}{W}$ and $\alpha = \frac{\lambda}{\lambda_{\max}}$; But $\lambda_{\max} = \frac{W}{mL}$ and thus the equation become:

$$T_h = \frac{L}{W} \frac{1}{1 - m\lambda \frac{L}{W}} \qquad \text{(Eq.4.19)}$$

where : L = data unit length [bytes], W = transmission rate [bytes/s], m = the number of the sensor in the sensor neighborhood and λ = the average arrival rate of the data unit at transmission.

When a data unit is generated it will pass from sensor to sensor up to the sink and will have at every node different delays that could be modeled by the equation 4.19. The data length unit and transmission rate are defined as primary parameters in the table 3.2 of paragraph 3.41 "Time Equation Parameters". It is mandatory to model the other two parameters: number of sensors in the neighborhood – m – and the average arrival rate – λ. As shown in the equation 3.1, the number of sensors in the neighborhood depends on the radio range radius. The average arrival rate depends on the average generation rate of the sensor – γ and an internal (transit) data unit arrival rate τ – a randomly parameter in our model.

4.4.3 Data Unit Generation and Representation

Assuming that a data unit is generated by the sensor number 3 in Fig. 4.1 This data unit will pass from the sensor 3 to the sensor 2, from the sensor 2 to the sensor 1 and from the sensor 1 to the sink. Each time when the data unit is transmitted from nodes 3, 2 and 1 toward the next node it meets different situations from one sensor to another. These situations will determine the delay at each node (delay per hop). The total travel time along the route is the sum of the delays at each hop. In this case, as sensor 3 generates a data unit, it is to determine three delays. To generalize, every data unit has the number of delays equal to the number of hops up to the sink. These delays have to be estimated using the mathematical model. But at each sensor the data unit meets different situations which could be modeled by the following parameters:

$$p_{i,h}^{(j)}(h^{(j)}, \rho^{h)}, m^{h)}, \tau_{i,h}^{(j)}, \lambda_{i,h}^{(j)}, T_{i,h}^{(j)}) \qquad \text{(Eq.4.20)}$$

In this notation, $p_{i,h}^{(j)}$ represents the data unit number *i* generated by sensor *j,* with $h^{(j)}$ = number of hops from the sensor *j* up to the sink, $\rho^{(h)}$ = route of the data unit from the sensor *j* in hop *h*, $m^{(h)}$ = number of sensor in the neighborhood from the sensor in hop *h*, $\tau_{i,h}^{(j)}$ = data unit internal rate, $\lambda_{i,h}^{(j)}$ = data unit arrival rate and $T_{i,h}^{(j)}$ = the delay of the *i* data unit generated by sensor *j* at the hop *h*.

If *j = 3,* as in the figure 4.1, it will be three representations for one generated data unit *i*:

$$p_{i,3}^{(3)}(3;\rho^{(3)} = 3,2,1,s;m^{(3)};\tau_{i,3}^{(3)};\lambda_{i,3}^{(3)};T_{i,3}^{(3)})$$
$$p_{i,2}^{(3)}(2;\rho^{(2)} = 2,1,s;m^{(2)};\tau_{i,2}^{(3)};\lambda_{i,2}^{(3)};T_{i,2}^{(3)})$$
$$p_{i,1}^{(3)}(1;\rho^{(1)} = 1,s;m^{(1)};\tau_{i,1}^{(3)};\lambda_{i,1}^{(3)};T_{i,1}^{(3)})$$

The total of route delay (which corresponds with the travel time) for the *"i"* data unit, generated at the sensor 3 will be:

$$T_{i,t}^{(3)} = \sum_{h=1}^{3} T_{i,h}^{(3)}$$

The notation is changed for the index because there are only the hop delays and therefore *h* will indicate the hop number and *i* will indicate the data unit identity. The exponent remains the same: sensor identity.

If sensor 2 generates a data unit it will have two representations:

$$p_{i,2}^{(2)}(2;\rho^{(2)} = 2,1,s;m^{(2)};\tau_{i,2}^{(2)};\lambda_{i,2}^{(2)};T_{i,2}^{(2)})$$
$$p_{i,1}^{(2)}(1;\rho^{(1)} = 1,s;m^{(1)};\tau_{i,1}^{(2)};\lambda_{i,1}^{(2)};T_{i,1}^{(2)})$$

In general, the data unit *i*, generated by a sensor *j* and a total of *H* hops will have a total of route delay (which corresponds with the travel time):

$$T_{i,t}^{(j)} = \sum_{h=1}^{H} T_{i,h}^{(j)} \qquad \text{(Eq.4.21)}$$

4.4.4 Parameters Modeling

In order to obtain the representations of each data unit along the path, we have to model each parameter in a representation given by equation 4.20.

In a field with dimensions X x Y it will be scattered in a random manner a number of N the sensors. Each sensor will have a position given by $Z_j(x_j;y_j)$ where

$$0 \le x_j \le X$$
$$0 \le y_j \le Y$$

Each coordinate of the sensor is obtained randomly by using a uniform distribution between 0 and X and respectively between 0 and Y.

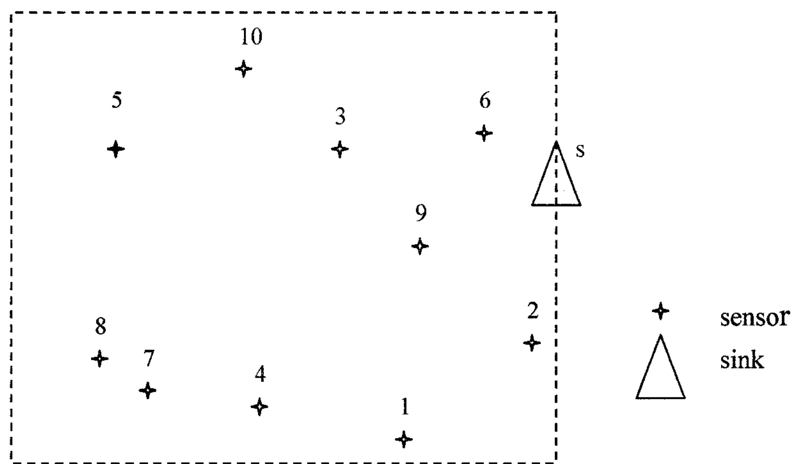


Figure 4.2 Sensors field representation

The sink node will be placed at the border of the field ($x_s = X$), but randomly on this border. Its coordinates will be $Z_s(X,y_s)$. As for a sensor, the coordinate y_s is obtain by uniform distribution between 0 and Y.

For 10 sensors, the field representation could be as in the figure 4.2 (X=Y):

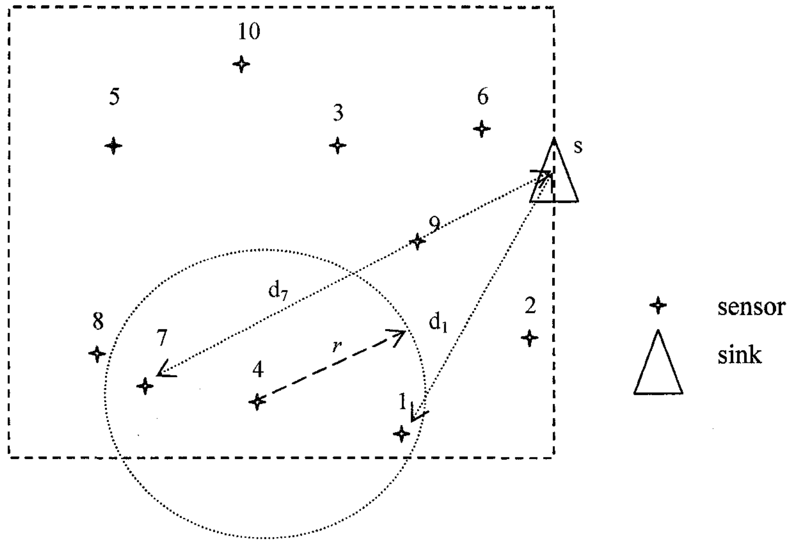


Figure 4.3 Routing candidates

If the radius radio range is r it will be possible to determine the number of sensors and the identity of the sensors in the neighborhood for each sensor. The number of sensors in the neighborhood represents the m parameter. The sensor identities in a sensor neighborhood give the sensor candidates for routing.

Table 4.1 Sensor route characteristics

sensor identity	**m**	**next hop**	**route**	**hops (H)**
1	3 (1, 2, 4)	2	1 → 2 → s	2
2	3 (2, 9, s)	s	2 → s	1
3	4 (3, 6, 9, 10)	6	3 →6 →s	2
4	3 (1, 4, 7)	1	4 → 1 → 2 → s	3
5	2 (5, 10)	10	5 → 10 →3 →6 →s	4
6	4 (3, 6, 9, s)	s	6 → s	1
7	3 (4, 7, 8)	4	7 → 4 → 1 → 2 → s	4
8	2 (7, 8,)	7	8 → 7 → 4 → 1 → 2 → s	5
9	5 (2, 3, 6, 9, s)	s	9 → s	1
10	3 (3, 5, 10)	3	10 → 3 →6 →s	3

A static routing will be considered which depends on the distance between the candidates and the sink, as is shown in the figure 4.3. The nearest candidate from the sink becomes the next sensor to forward the data units.

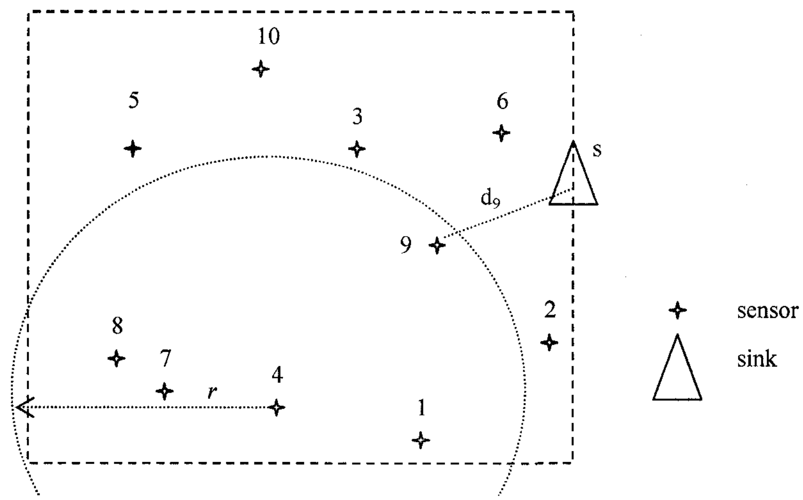


Figure 4.4 Routing candidates for a greater radio range radius

The sensor number 4 has $m = 3$ and the two candidates for data unit forwarding which are 7 and 1. As the distances between each candidate and the sink are d_1 and d_7 and because d_7 is greater than d_1, the data units from sensor 4 will be forwarded towards the sensor 1.

When the next step from each sensor it is known, is possible to obtain the route from each sensor and the number of hops towards the sink as in the table 4.1

Route characteristics change when radio range radius changes. In the figure 4.4, it is kept the same sensor deployment but it is used a greater radio range radius. The route characteristics change as in the table 4.2.

As it can be quickly seen, when the radio range radius increases, the number of hops on the path decreases, but the number of sensors in a neighborhood increases.

Table 4.2 Sensor routes characteristics for a greater radio range radius

sensor identity	m	next hop	route	hops (H)
1	4(1, 2, 4,9)	9	$1 \rightarrow 9 \rightarrow s$	2
2	5(1, 5, 6, 9, s)	s	$2 \rightarrow s$	1
3	7 (3, 5, 6, 7, 9, 10, s)	s	$3 \rightarrow s$	1
4	5 (1, 4, 7, 8, 9)	9	$4 \rightarrow 9 \rightarrow s$	2
5	5 (3, 5, 7, 8, 10)	3	$5 \rightarrow 3 \rightarrow s$	2
6	6 (2, 3, 6, 9, 10, s)	s	$6 \rightarrow s$	1
7	6 (3, 4, 5, 7, 8, 9)	9	$7 \rightarrow 9 \rightarrow s$	2
8	4 (4, 5, 7, 8)	4	$8 \rightarrow 4 \rightarrow 9 \rightarrow s$	3
9	8 (1, 2, 3, 4, 6, 7, 9, s)	s	$9 \rightarrow s$	1
10	6 (3, 5, 6, 7, 9, 10)	6	$10 \rightarrow 6 \rightarrow s$	2

Given the scenario from the figure 4.3 and the routes characteristics in table 4.1, if each sensor generates data units, these data units will have different representations, depending on the hop along the route up to the sink. Assuming that sensor 8 and sensor 10 generate a data unit (with its identity *i*), each data unit will have the following representation. Sensor 8 data unit representation is:

$$p_{i,5}^{(8)}(5; \rho^{(5)} = 8,7,4,1,2,s; m^{(5)} = 3; \tau_{i,5}^{(8)}; \lambda_{i,5}^{(8)}; T_{i,5}^{(8)}))$$
$$p_{i,4}^{(8)}(4; \rho^{(4)} = 7,4,1,2,s; m^{(4)} = 3; \tau_{i,4}^{(8)}; \lambda_{i,4}^{(8)}; T_{i,4}^{(8)}))$$
$$p_{i,3}^{(8)}(3; \rho^{(3)} = 4,1,2,s; m^{(3)} = 3; \tau_{i,3}^{(8)}; \lambda_{i,3}^{(8)}; T_{i,3}^{(8)}))$$
$$p_{i,2}^{(8)}(2; \rho^{(2)} = 1,2,s; m^{(2)} = 3; \tau_{i,2}^{(8)}; \lambda_{i,2}^{(8)}; T_{i,2}^{(8)}))$$
$$p_{i,1}^{(8)}(1; \rho^{(1)} = 2,s; m^{(1)} = 3; \tau_{i,1}^{(8)}; \lambda_{i,1}^{(8)}; T_{i,1}^{(8)}))$$

Sensor 10 data unit representation is:

$$p_{i,3}^{(10)}(3;\rho^{(3)}=10,3,6,s;m^{(3)}=3;\tau_{i,3}^{(10)};\lambda_{i,3}^{(10)};T_{i,3}^{(10)}))$$
$$p_{i,2}^{(10)}(2;\rho^{(2)}=3,6,s;m^{(2)}=4;\tau_{i,2}^{(10)};\lambda_{i,2}^{(10)};T_{i,2}^{(10)}))$$
$$p_{i,1}^{(10)}(1;\rho^{(1)}=6,s;m^{(1)}=4;\tau_{i,1}^{(10)};\lambda_{i,1}^{(10)};T_{i,1}^{(10)}))$$

The route total hops delay for the data unit number i sent from the two sensors is:

$$T_{i,t}^{(8)}=T_{i,5}^{(8)}+T_{i,4}^{(8)}+T_{i,3}^{(8)}+T_{i,2}^{(8)}+T_{i,1}^{(8)}$$

$$T_{i,t}^{(10)}=T_{i,3}^{(10)}+T_{i,2}^{(10)}+T_{i,1}^{(10)}$$

In order to obtain each $T_{i,h}^{(j)}$ value it is necessary to build the function $\tau_{i,h}^{(j)}$ which represents an internal data arrival rate. Its value that has to be added at the generation rate to obtain the arrival rate at each sensor, for the hop h and data unit i.

$$\lambda=\gamma+\tau \qquad \text{(Eq.4.21)}$$

where λ is the arrival rate data units (data units that have to be transmitted) by sensor, γ is the generation rate data units of the sensor (the same for all sensors), and τ models the data units internal rate that exist in a sensor. This function has a random value of a uniform distribution on the interval $(0,\lambda_{\max})$:

$$\tau=\lambda_{\max}\cdot rand(0,1) \qquad \text{(Eq.4.22)}$$

The maximum value of $\lambda_{\max}$ is given in the equation 3.8: $\lambda_{\max}=\dfrac{W}{mL}$. This value is determined for entire network because m is the number of the sensor nodes in the neighborhood for entire network (it is an average). Based on the equation 4.21, 4.22 and 3.8 it is possible to calculate the data arrival rate for every data unit per hop.

Protocol Data Arrival Rates

In order to make the difference between the two protocols the following value of the data arrival rate will be taken:

1. for the protocol UDP, there is no any control of the data arrival rate and therefore the value of the λ is given by the equation 4.22

2. for BVP protocol, the traffic shaping mechanism limit the data arrival rate at $\lambda_{\lim}$ therefore if λ calculated by the equation 4.22 is greater then $\lambda_{\lim}$, in the delay calculation there will be used the value of $\lambda_{\lim}$. If the value of λ is less than $\lambda_{\lim}$, the delay will calculated with the value of λ [$\lambda := \min(\lambda_{\lim}, \lambda);$].

In the case of UDP protocol, if the data arrival rate is greater than the $\lambda_{\max}$, ($\lambda_{\max} = \dfrac{W}{mL}$) the delay calculated by the equation 4.19 become negative. In this case a large value will be chosen to be assigned to the delay instead of the calculated value.

4.4.5 The Model Limits

The model built for the protocol UDP and BVT comparison has the following limitations:

- the function τ which could not model exactly the real situation
- the delay per hop calculation is based on the assumption that all the nodes have the same data unit arrival rate
- the assumption that in a sensor network the routing is static
- BVP alleviates the network because it discards all the data units that have the travel time greater than the deadline, which is not modeled
- the data unit discard in the case of the UDP protocol (when the buffers are full) are not modeled

- processing time at the transport layer is neglected
- the scheduling mechanism for BVP is not modeled

4.4.6 Methodology Implementation for MATLAB

In order to arrive at the comparison results given by MATLAB, here are the steps that have to be followed:

1. sensor position generations on the field, as in the figure 4.2; values to choose: *N, X,Y;* obtained values: sensor positions $j(x_j, y_j)$; $1 \le j \le N$

2. sink position generation on the field $s(X, y_s)$;

3. based on the value of the radius radio range: *r,* calculation for each sensor: the number of hops, route and number of sensors in the neighborhood; value to choose: *r;* obtained values: $h^{(j)}$, $\rho^{(j)}$ and $m^{(j)}$

4 data units generation; value to choose *P* which is the number of data units generated by a sensor; obtained value *n,* the total number of data unit generated by the network

5. data unit representation based on the number of hops, and routes

6. random values generation for all representations: rand (0,1)

7. values of τ function computation: $\tau_{i,h}^{(j)} (1 \le i \le P, 1 \le j \le N, 1 \le h \le H)$

8. values of λ function computation for each data unit representation and for the two protocols

9. values of delay computation for every representation: $T_{i,h}^{(j)}$ and values of the travel delay for all hops along the route for the two protocols

10. values of the travel time for each data unit and for the two protocols

11. values of the numbers of data units that are in time $n_{T\leq D,U}; n_{T\leq D,B}$ for the two protocols and thus the approximation of the timelyputs

13. protocols comparison based on the value obtained at points 10 and 11.

4.4.7 Model Implementation

The implementation of the mathematical model for protocol UDP versus BVP comparison was made with MATLAB 7.1 software using a computer having a CPU Pentium 4, 3.00 GHz, and 1GB RAM. The computer operating system is Microsoft Windows XP.

The implementation was made with three different programs: i)a program for sensor network generation, ii) a program for timelyput comparison and ii) a program for travel time comparison.

Sensor Network Configuration

The network was modeled by a program that implements a mathematical model of a flexible sensor network and calculates the routes and the number of hops from each sensor toward the sink. As input data there are: the number of sensor nodes: *N*, the number of sink nodes: *S*, the radio range radius: *r*, and the field dimensions: *X* and *Y*. All the input data can be changed in order to obtain different sensor networks

The program scatters randomly the sensor and sink nodes over the field. Based on the static routing, explained in the paragraph 4.4.4 "Parameters Modeling" and the radio range radius, it determines for every sensor the number of sensors in its neighborhood, the route (sensor identities) and the number of hops up to the sink. The program also calculates values for entire network: number of sensors in the neighborhood, maximal data arrival rate and maximal data generation rate per sensor. The program also indicates the percentage of the "successful sensors" (whose data units arrive at a sink) and the percentage of the "unsuccessful sensors". These former sensors are divided into: "isolated sensors" (sensor that cannot forward its data units), "blocked sensors" (sensors that can forward its data units but these data units arrive, eventually, at an isolated sensor) and "looped sensors" (sensors whose data units circle in a loop and will never arrive at a sink).

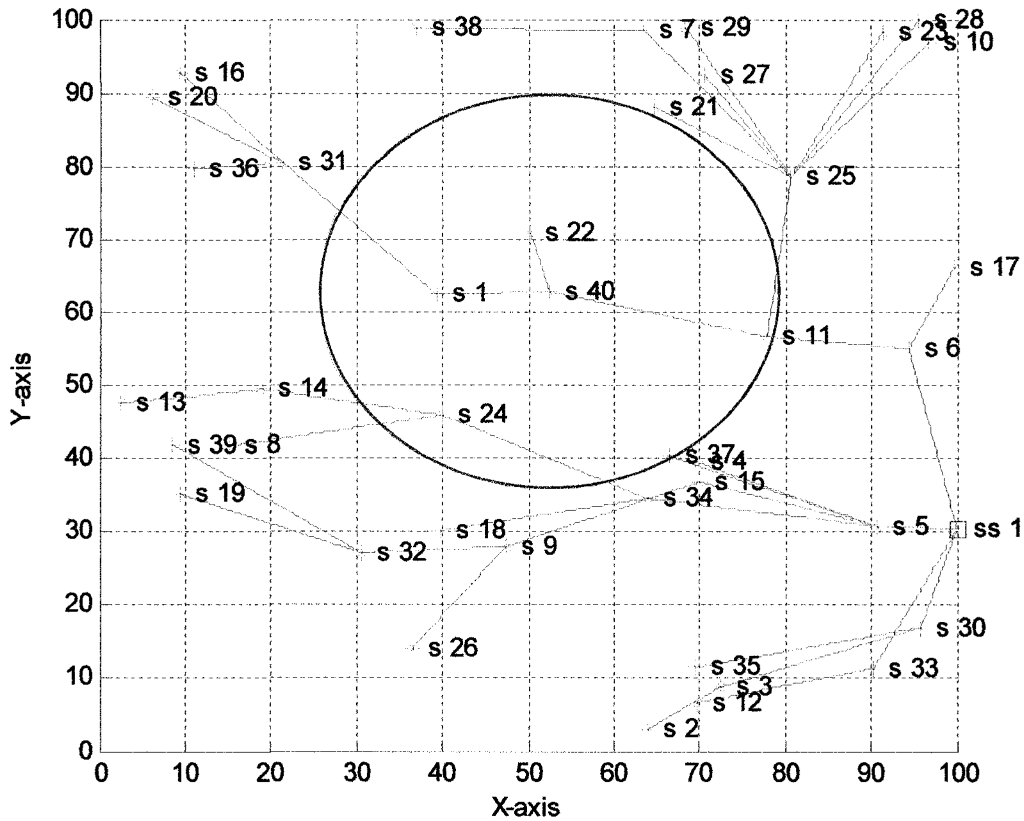


Figure 4.5 Sensors and sink positioning in the field for a tiny network

The random numbers for nodes positioning are generated using the function "rand" which has the value of "state" as parameter. This parameter could be changed in order to obtain a different position of the nodes in the field when all input data are kept at the same value.

As an example, in the figure 4.5 a tiny sensor network is presented with the following input data: $N = 40$, $S = 1$, $r = 27$ m, $X = 100$ m, $Y = 100$ m, state = 100. The number of the successful sensors is 40 (all sensor data units have a path to arrive at the sink). A sensor is represented by a red character "*" and the name of the sensor (sensor identity) is given next to the sensor. The sink node is represented by a black empty square shape identified by "ss1". The blue lines between sensors indicate the routes of the data units. As a data unit will go towards a sink always positioned on the right side of the field, a blue line indicates that the left side line sensor forwards the data units towards the right side line sensor. A sensor will forward its data units and all the data units of the sensors from the upstream. Finally, all the routes converge to the sink nodes. For the sensor 16, the route is: s16, s31, s1, s40, s11, s6 and ss1, and it has 6 hops. For the sensor "s40" was drawn the cover limit of the radio range of its communication system (green). This shows the number of sensors in the sensor neighborhood (in this case 6: s1, s11, s22, s24, s37, s40) and the best candidate to forward the data units. "s40" will forward the data units to "s11", as can be seen on the figure 4.5.

Timelyput Comparison

The implementation for timelyput comparison of the two protocols BVP and UDP was made by a program with two variants: one variant makes a comparison in two dimensions: timlyput versus deadline with a fixed value of data unit generation rate and the second variant makes a comparison in three dimensions: timelyput versus deadline and data units' generation rate. The data input for these programs are the network configuration given by the program that models the sensor network, number of data units generated by each sensor *P*, the data unit length *L*, the transmission rate *W*. The programs generate, for every sensor, a matrix with *H*P* lines where *H* is the number of hops from that sensor to the sink. The number of lines of this matrix stands for all data unit representations for a single sensor. On the first column we mark the sensor identities which are encountered by the data unit along the route (hops) while in the second column is generated a random vector which represents the values of data unit internal rate: τ . The third and fourth columns represent the delays per hop for UDP and BVP respectively. In order to have a consistent comparison the same value of the data unit internal rate τ is used for both protocols. This data unit internal rate is added to the data unit generated rate and the data unit arrival rate λ is obtained. Based on the data unit arrival rate, the network configuration and input data it is calculated the hop delay time for each representation. For BVP protocol, the delay hop is calculated taking into account that the protocol limits the value of the data unit arrival at a maximum accepted value. The value of the travel time for each data unit is obtained by adding all the hop delays of all hops along the route of the data unit. For every protocol the travel time per data unit and per sensor are put into different matrixes with N lines and P columns.

The first variant of the program makes a comparison of protocols in two dimensions. If a deadline D is established, the program evaluates the number of data units that arrived in time, thus it evaluates the protocols timelyput. If the deadline takes a range of values the calculated timelyput for each protocol will be represented as a graphic. The results of this program are the figures: Fig 4.7, 4.8, 4.13, 4.14. In these figures the BVP timelyput is drawn in red and the UDP timelyput in blue. On the x-axis the deadline takes different ranges of values depending on the transmission rate W.

The second variant of the program generates protocols timelyput surfaces versus deadline and data units generated rate. A vector with equidistant values, of the data unit generation rate, is generated. For every value it is calculated the timelyput as it is described above. The results are put in two matrixes with three dimensions and visualized as surfaces. For each evaluated case were obtained three figures: BVP timelyput surfaces in Fig. 4.9 and 4.15, UDP timelyput surfaces in Fig. 4.10 and 4.16 and the timelyput difference surfaces between the BVP and UDP in Fig. 4.11 and 4.17.

Travel Time Comparison

The implementation of the travel time comparison was made with a program which draws two histograms (one for each protocol) that indicate the numbers of data units that arrive in an interval of time. The input data for this programs are two matrixes with the data units travel times obtained with the first variant of the program for timelyput comparison. The histograms for the studied cases are shown in the figures: Fig. 4.12 and 4.18.

4.4.8 Results Presentation

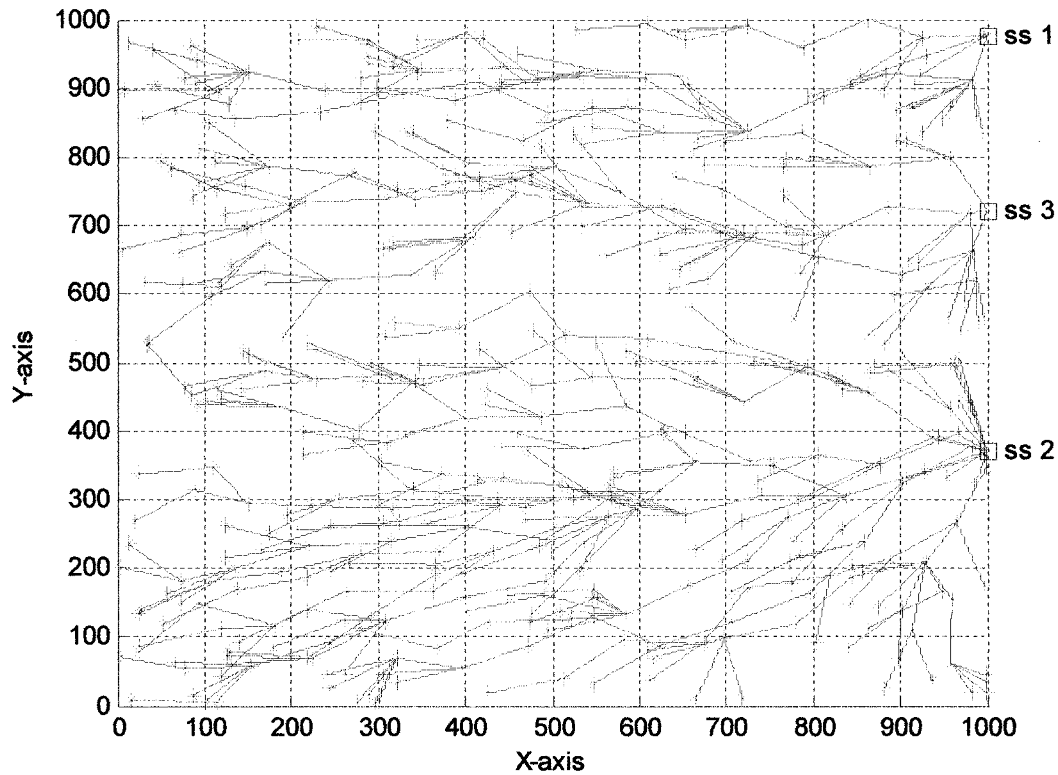


Figure 4.6 Sensors and sinks positioning in the field for scenarios network

The protocols comparison was made on a sensor network presented in the Fig. 4.6 which has $N = 500$ sensor nodes and $S = 3$ sink nodes scattered on a surface of $X =$ 1000 m X $Y = 1000$ m. The radio range radius is $r = 100$ m. All sensors on the network have a route towards one sink (all sensors are successful). For the clarity, in the figure, we omitted the nodes identity. The number of data units generated by each sensor is $P =$ 10000. The length of data unit is 128 bytes. There were taken, for comparison two scenarios, depending of the transmission rate capacity: a) low transmission rate $W = 64$ kbit/s (8 kbytes/s); b) high transmission rate $W = 2048$ kbit/s (256 kbyte/s). There are also the results for the third scenario for a middle transmission rate $W = 512$ kbit/s (64 kbyte/s) which are presented in the Annex 2.

Sensor Network with Low Transmission Rate

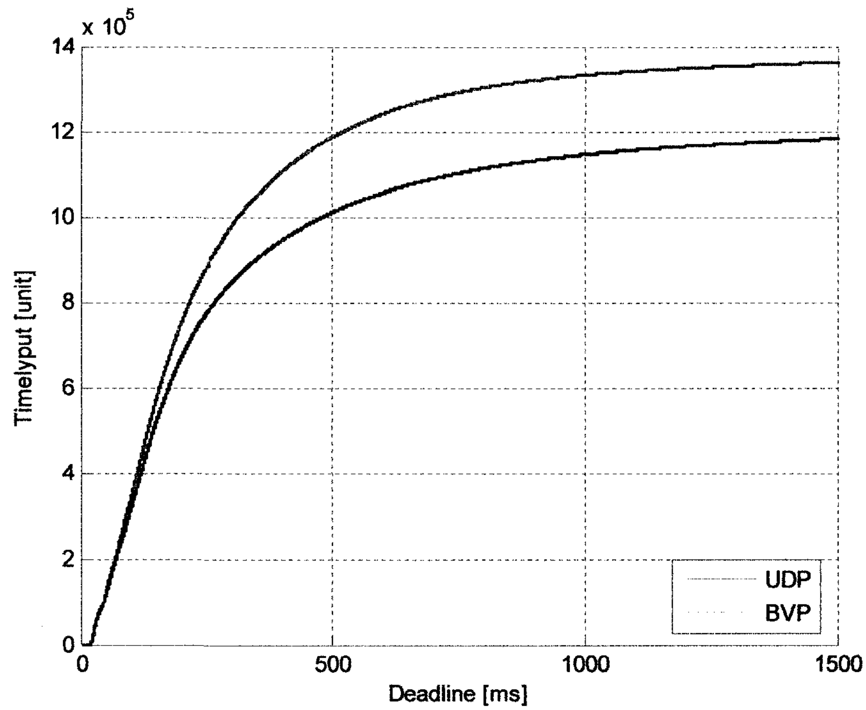


Figure 4.7 Protocol timelyputs versus a large deadline interval for low transmission rate

In the figure 4.7 and 4.8 the two protocols are compared for the timelyput. In the figure 4.7 the deadline range is the interval 0 – 1500 ms. Data unit generation rate is at a constant value of 1.125 data units/s. The figure 4.8 shows better the first part of the interval because the deadline is in the interval range of 0 – 600 ms. At the beginning of the deadline interval 0 – 20 ms there is no timelyput for both protocols. This is normal because the data unit transmission time is $T_o = \frac{L}{W}$ 16 ms (W in kbyte/s). After the deadline of 20 ms the two protocols have a liner rise of the timelyput.

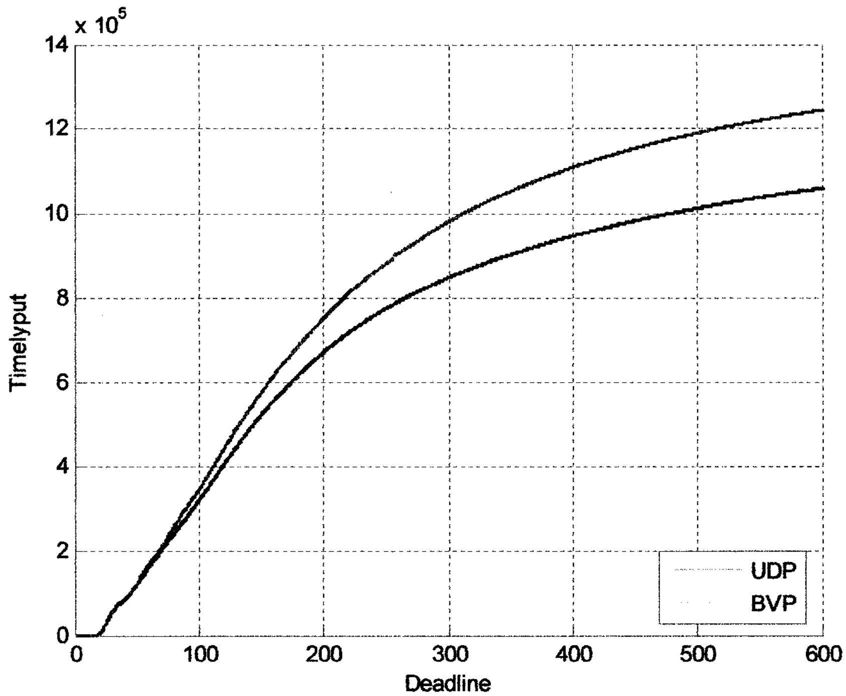


Figure 4.8 Protocol timelyputs versus a short deadline interval for low transmission rate

The difference between protocols is very little up to the deadline value of 80 ms. At this value the timelyput protocols is about 200,000 units. Increasing the deadline, the protocols became to differentiate. From the value of 200 ms, the timelyput protocols start to rise in a non linear manner. After the value of the 600 ms the increase of the timelyput is insignificant. At the end of this interval the number of data units that arrive before the deadline is almost constant. Starting with the deadline value of 600 ms, from the total number of the data units generated by the network (5,000,000) only about 1,100,000 data units arrive in time for UDP protocol that represents a percentage of 22% and about 1,300,000 data units arrive in time for BVP protocol that represents 26% of the generated data units. The Bonaventura protocol has a timelyput better than UDP with about 200,000 units that represent 4% from the total number of the data units and an increase with 18% than the UDP protocol.

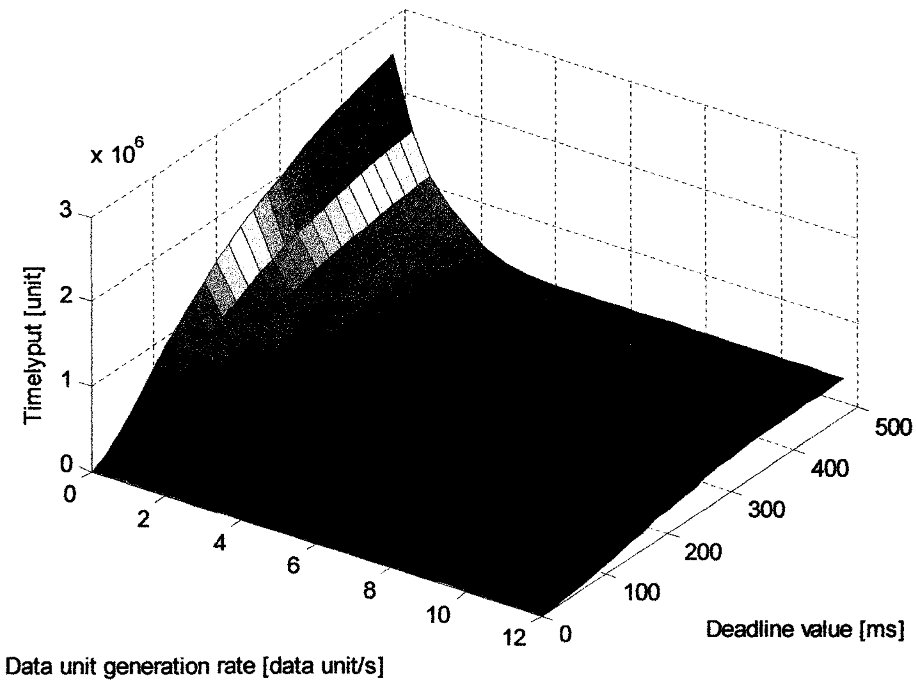


Figure 4.9 BVP timelyput surface for low transmission rate

In the figures 4.9, 4.10 and 4.11 are drawn the surfaces of the variation of the timelyput versus deadline and data unit generation rate for Bonaventura protocol (fig. 4.9), UDP (fig. 4.10) and the timelyput difference between the two protocols. The deadline interval is 0 – 500 ms and the data units generation rate interval is 0 – 10 data units/s. When the data unit generation rate is very low and the deadline value is large, the timelyput of the two protocols is at maximum possible: 2,500,000 units which correspond at 50% of the maximum throughput of the network. As the data generation rate increases, the timelyput of the two protocols decreases, but the difference between protocols increases as can be seen on the figure 4.11. When the data unit generation rate arrives at the value of 5 data units/s the network is in a total congestion and the UDP protocol cannot carry any data unit with a travel time less than 500 ms. Under the same conditions, BVP protocol has a timelyput of 400,000 units which represent 8% of the total generation number of data units. When the deadline value decreases, the timelyput difference decreases in a linear manner and for very little values of deadline the difference is zero.

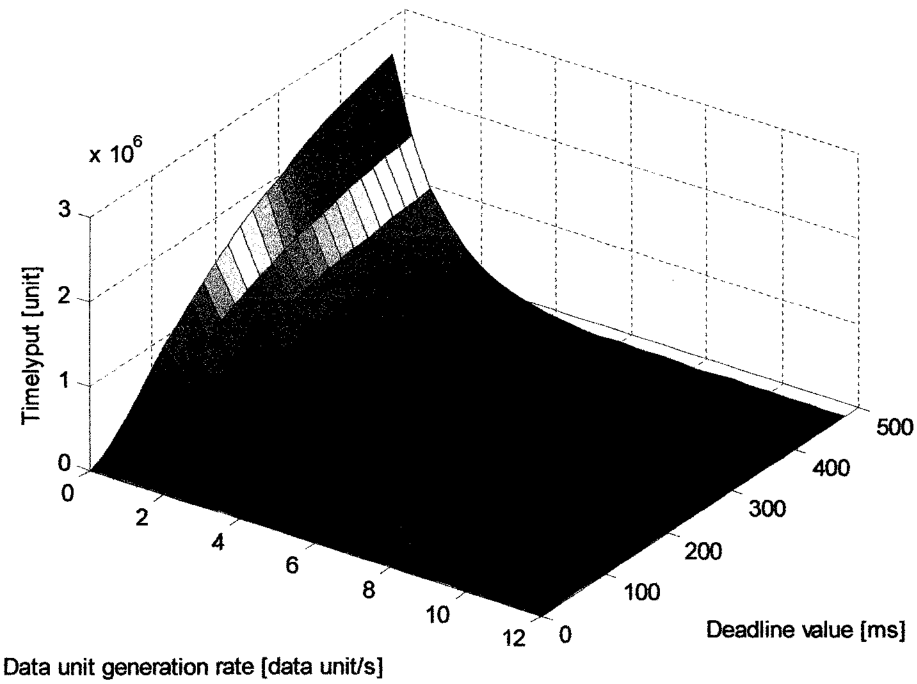


Figure 4.10 UDP timelyput surface for low transmission rate

In the figure 4.12, are presented two histograms one for BVP protocol and one for UDP protocol in order to compare the two protocols for the travel time of the data units. In this case no deadline is imposed for the travel time. The histograms take the interval 0 – 1000 ms and divide it in interval of 20 ms. For each interval, the number of the data units are counted. As it can be seen, the number of data units in each interval is greater for BVP than UDP. The most important difference between these protocols is given in the interval 100 – 140 ms.

As conclusion, the transmission rate capacity W = 64 kbit/s (16 kbyte/s) determines long travel times for data units (as the transmission time for a single data unit is 16 ms) and it also imposes little value of the data unit generation rate (calculated

γ_{max} = 0.375 data unit/s). At a data unit generation rate of 5 data units/s the network is in total congestion.

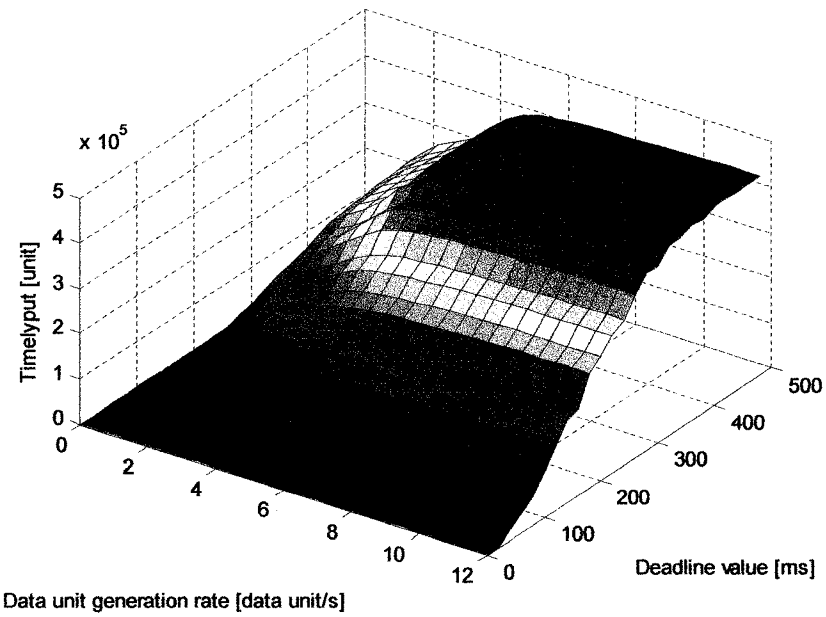


Figure 4.11 Timelyput difference surface for low transmission rate

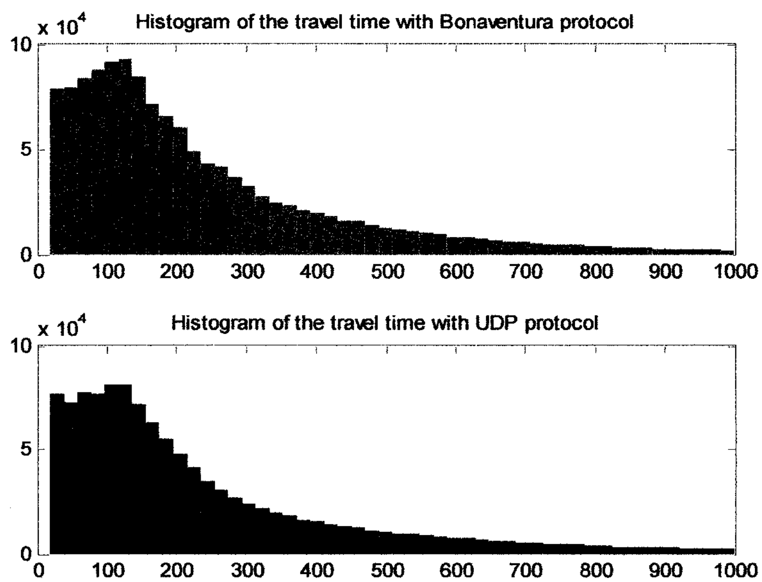


Fig. 4.12 Histograms for data units travel times for low transmission rate

The difference between the two protocols is little when the network resources are not exceeded, but this difference increases when the network is used at full capacity.

Sensor Network with High Transmission Rate

There are five figures for timelyput comparison. In the figure 4.13 the deadline range is the interval 0 – 60 ms. Data unit generation rate is at a constant value $\gamma = 36$ data unit/s. The figure 4.14 shows the timelyput for the first part of the deadline interval (0 – 15 ms). Because the transmission time of a single data unit $T_o = \frac{L}{W}$ is 0.5 ms (W = 256 kbyte/s), at the beginning of the deadline interval 0 – 1 ms there is no timelyput for both protocols. After the value of 2 ms of deadline the two protocols have a liner rise of the timelyput.

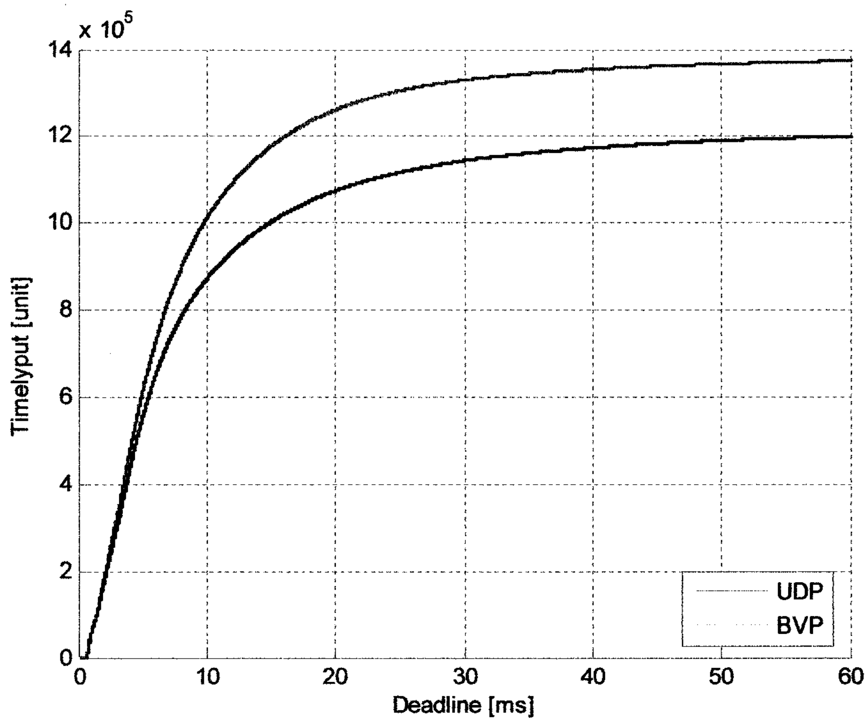


Figure 4.13 Protocol timelyputs versus a large deadline interval for high transmission rate

The difference between protocols is very little up to the deadline of 10 ms. Increasing the deadline, the protocols become to differentiate. At about the value of 10 ms the timelyput protocols starts to rise in a non linear manner. After the value of the 40 ms the increase of the timelyput is insignificant. At the end of the deadline interval, the number of data units that arrive before the deadline is almost constant. Starting with the deadline value of 40 ms, from the total number of the data units generated by the network (5,000,000) only about 1,180,000 data units arrive in time for UDP protocol which represents a percentage of 24% and about

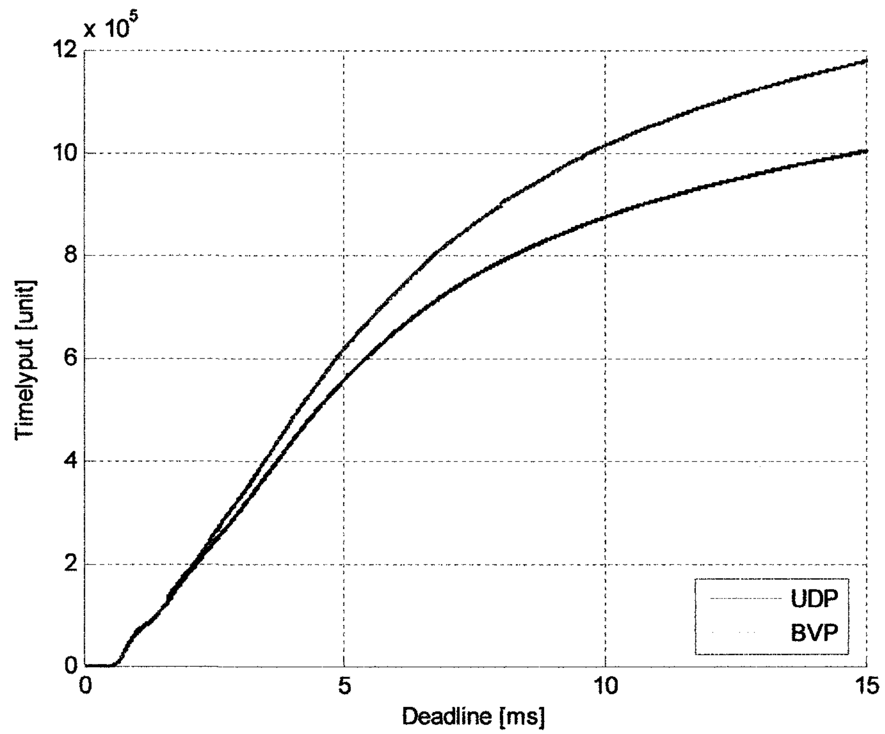


Figure 4.14 Protocol timelyputs versus a short deadline interval for high transmission rate

1,380,000 data units arrive in time for BVP protocol that represents 28% of the generated data units. The Bonaventura protocol has a timelyput better than UDP with about 200,000 data units that represent 4% from the total number of the data units and an increase with 17% than the UDP protocol.

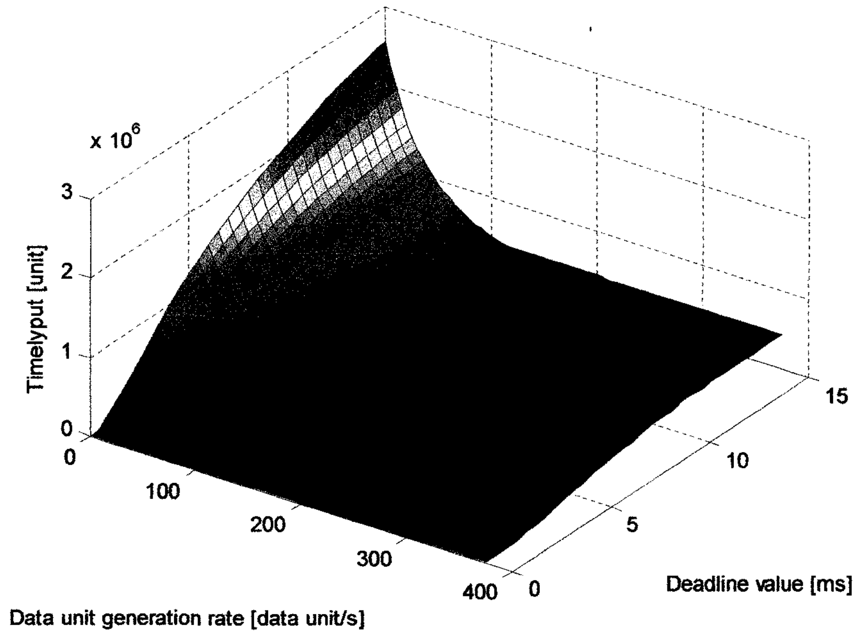


Figure 4.15 BVP timelyput surface for high transmission rate

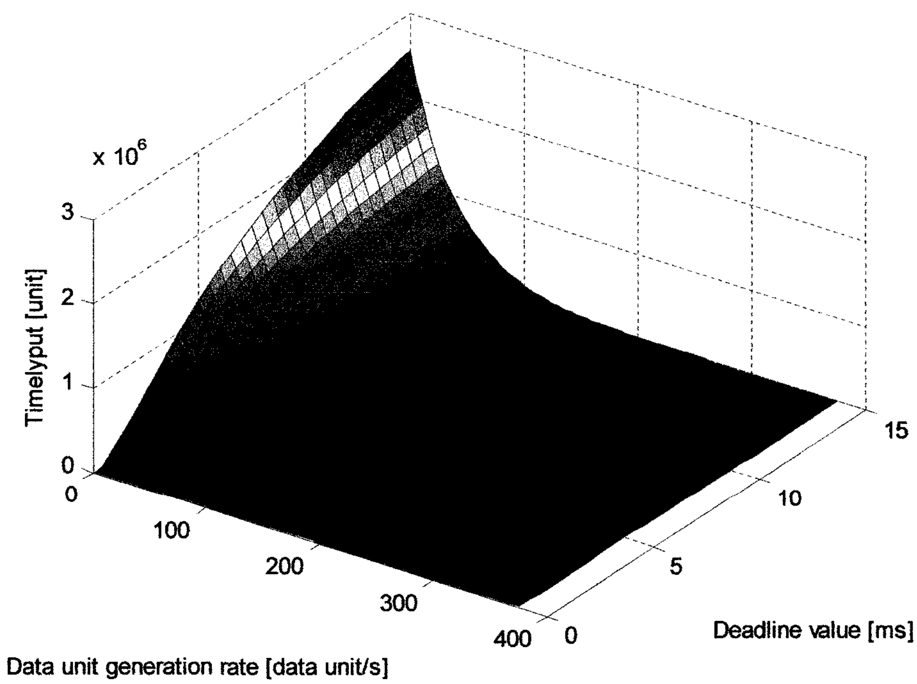


Figure 4.16 UDP timelyput surface for high transmission rate

In the figures 4.15, 4.16 and 4.17 are drawn the surface of the variation of the timelyput versus deadline and data unit generation rate for Bonaventura protocol (fig. 4.15), UDP protocol (fig. 4.16) and the timelyput difference between the two protocols (fig 4.17). The deadline interval is 0 – 15 ms and the data units generation rate interval is 0 – 250 data units/s. When the deadline value is large (about 15 ms) and the data unit generation rate is very low, the two protocols have almost the same behavior: the timelyput is about 2,500,000 units which correspond at 50% of the total throughput of the network. As the data generation rate increases, the timelyput of the two protocols decreases, but the difference between protocols increases as can be seen on the figure 4.17. When the data unit generation rate arrives at the value of 150 data units/s the network is in a total congestion and the UDP protocol cannot carry any data unit with a travel time less than 15 ms. In the these conditions, BVP protocol has a timelyput of 400,000 units which represent 8% of the total generation number of data units. When the deadline value decreases, the timelyput difference decreases in a linear manner and for very little values of deadline the difference is zero.

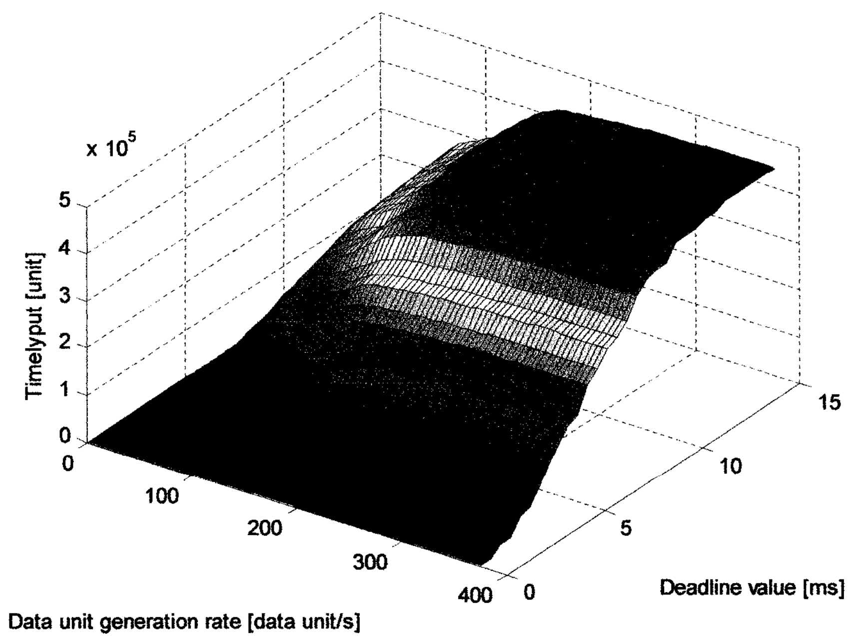


Figure 4.17 Timelyput difference surface for high transmission rate

Figure 4.18, shows two histograms to compare data units travel time of the two protocols. The histograms take the interval 0 – 50 ms and divide it in interval of 1 ms. For each interval are counted the number of the data units. As it can be seen the number of data units in each interval is greater for BVP than UDP. The most important difference between these protocols is in the interval 2 – 5 ms.

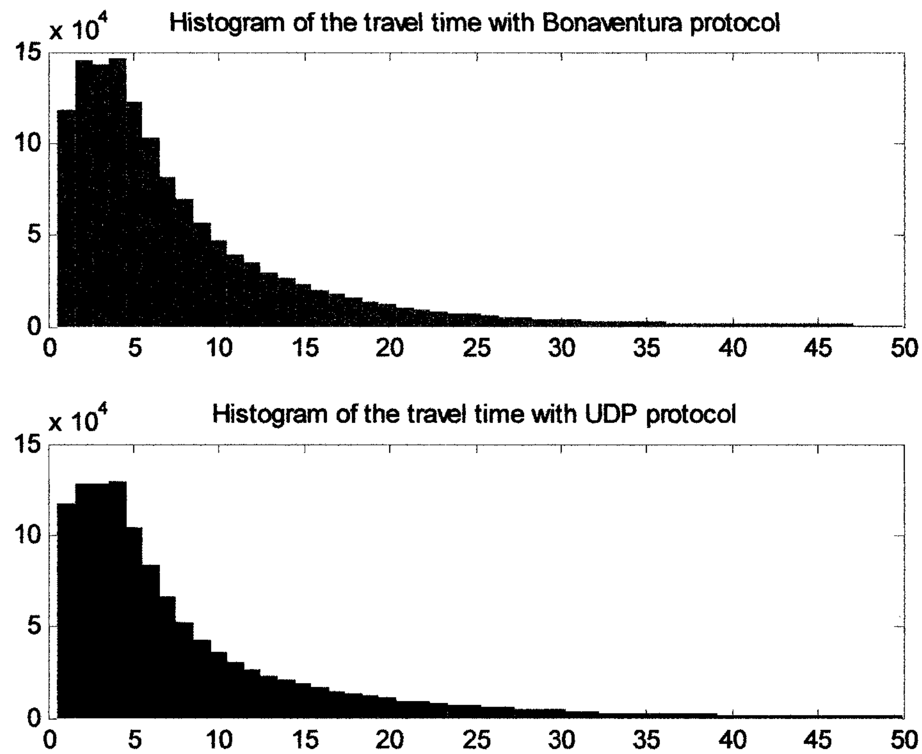


Figure 4.18 Histograms for data units travel times for high transmission rate

As a conclusion, the transmission rate capacity $W = 2048$ kbit/s determines very short travel times for data units (because the transmission time for a single data unit is 0.5 ms) and it also allows large values of the data unit generation rate. At a data unit generation rate of 150 data units/s the network is in total congestion. The difference between the two protocols is little when the network resources are not overwhelmed, but this difference increases when the network is used at full capacity.

Chapter 5 Conclusion

This work presents a transport protocol for a real time communication that is suitable in the wireless sensor networks. In the specialized literature, for the network and data-link layer, there are presented some real-time protocols for wireless sensor network, as there are described in the chapter two of this work, but for the transport layer there is no any real-time communication protocol. The existing TCP and UDP protocols are not suitable for real-time communications and for wireless technology, but in this case UDP behaves better than TCP. As a consequence, the performance estimation of the designed protocol was compared with UDP protocol.

The designed protocol – Bonaventura (BVP) – tries to change the vision of a transport protocol as process that runs only at the ends of the communication: the source and the destination. The decision in order to manage the time has to be taken locally in every node along the path and not globally from end to end as a classic transport protocol acts. There fore it is impossible to manage the time with only two entities. The BVP has entities that run in every node along the path and keep the control over the time for every data unit. These entities along the routes evaluate data units against the real-time communication requirements. As a consequence, the entities establish a new order of transmission of the data units towards the next node to support the data units that are late. The protocol BVP was compared with the protocol UDP using models implemented in MATLAB. The following conclusions could be drawn:

1. For a real-time and a wireless communication a transport has to be implemented at every node along the path. The transport protocol entity manages the time of each data unit in order that the data unit would meet the requirements of the real-time communication.

2. The designed protocol – Bonaventura – could be used for wireless sensor networks and for wireless sensor actor networks. The presence of actors will diminish the travel time but for design purpose it has to be taken into consideration the travel time up to the sink. The actors are mobile and this make impossible the guarantee of the

actors presence in the sensor vicinity and thus the communication between sensors and actors is made via sink nodes.

2. The scheduling filtering mechanism is one of the two protocol mechanisms. That mechanism is included in order to manage the time of the data units along the path. This mechanism evaluates each data unit in two dimensions: time and space. A priority function was developed in order to evaluate the data units and in order to assign a priority to data unit. Based on the priority, the data unit sending order, to the next node, is changed to support the data units that are late.

3. The traffic shaping mechanism is the second mechanism of the protocol. That mechanism is included in order to manage the characteristics of the sensor networks. It tries to limit the data unit arrival rate of each sensor. A sensor could generate easily a large amount of data units that are impossible to be carried by the network. This mechanism will keep the network under the limit of congestion. A data unit arrival limit has been obtained for this mechanism, but the implemented model has shown that this limit is more flexible and could be exceeded.

4. The cost to have a real-time communication for a wireless sensor network is time. The implementing of an entity in every node and having the two described mechanisms in every entity lead to a more complex protocol than UDP. Moreover, along the path, BVP will introduce processing times and will delay the data units. This could raise a paradox: in order to be in time, a data unit has to be delayed! As the waiting time in queues and transmission time of the data units are larger than the processing times, BVP will behave better than a classical transport protocol.

5. The mathematical comparison between the BVP and UDP protocols shows that:

- when the sensor network resources are not used at full capacity, there is little difference in the two protocols behavior, but when the resources are used at their limits (as in a sensor network) the BVP is very advantageous. Even if the sensors do not generate data units over the congestion level of the network, the randomly deployment of sensors could lead to congestion zones. In this case BVP eliminates these situations but UDP does not.

- the throughput of the data unit that arrive in time (timelyput) is better for the BVP with up to 18% than UDP.

- the model limits used to compare the protocols lead to almost the same behavior of the protocols using different transmission rate on the same sensor network configuration.

6. The model built for the protocol UDP and BVT comparison has limitations that could modify the obtained results. The main limitations are:

- the delay per hop calculation is based on a very simple model (M/M/1) of the neighborhood virtual queue and on the assumption that all sensors have the same data unit arrival rate
- processing time at the transport layer is neglected
- the scheduling mechanism for BVP is not modeled

7. Further works could be made in order to improve Bonaventura protocol. Here are some improvements:

- the protocol could determine the data unit generation limit dynamically, because this rate depends on the local sensor density. This will permit the sensors in an area with low density to generate more data units than the sensors in a local area with high density.

- the protocol could change dynamically the limit of the traffic shaping mechanism. This improvement could be done using the cross layer design in order that BVP protocol should be notified with the status of the physical layer. When the physical layer permits, the shaping mechanism could increase the limit and more data unit could be send by the sensor in a period of time.

- the protocol could modify the limit of the data unit arrival rate by changing the radio range radius. The data unit arrival could be increased by decreasing of the radio range radius. This may allow a sensor to send more data units and could be used when a sensor enters in a low level of congestion. The radio range decreasing will increase the hops number thus a greater travel time of the data units.

- depending on the application, the protocol could evaluate a data unit (for priority value) not only by time and space parameters. Other metrics could be added for evaluation purpose.

8. In order to improve the performance evaluation of the Bonaventura protocol, it could be simulated using the software NS2. (Network Simulator 2) or it could be implemented on real sensor network.

Bibliography

[1] STANCOVIC A. John, ABDELZAHER F. Tarek, LU Chenyang, SHA Lui, HOU C. Jennifer, "Real-Time Communication and Coordination in Embedded Sensor networks", Proceeding of the IEEE, Vol. 91, No. 7 July 2003, Pages 1002-10022

[2] CACCAMO Marco, ZANG Y. Lynn, SHA Lui, BUTTAZZO Giorgio, "An Implicit Prioritized Access Protocol for Wireless Sensor Networks", Proceeding of the 23-rd IEEE Real-Time Systems Symposium, December 2002, Pages 39-48

[3] CRENSHAW L. Tanya, TIRUMALA Ajay, HOKE Spencer, CACCAMO Marco, "A Robust Implicit Access Protocol for Real-Time Wireless Collaboration, Real-Time Systems, 2005 (ECRTS 2005) Proceedings 17th Euromicro Conference on Volume, Issue, 6-8 July 2005 Pages: 177 - 186

[4] AKKAYA Kemal, YOUNIS Mohamed, "A Survey on Routing Protocols for Wireless Sensor Networks", Elsevier Computer Science, Ad-Hoc Networks, Volume 3, Issue 3, 2005, Pages 325-349

[5] INTANAGONWIWAT Chalermek, GOVIDAN Ramesh, ESTRIN Deborah, HEIDERMANN, John, SILVA, Fabio, "Directed Diffusion for Wireless Sensor Networking", IEEE/ACM Transactions on Networking, Volume 11, Issue 1, February 2003, Pages 2-16

[6] KRISHNAMACHARI Bhaskar, ESTRIN Deborah, WICKER Stephen, "The Impact of Data Aggregation in Wireless Sensor Networks", Distributed Computing Systems Workshops, 2002 Proceedings 22nd International Conference on Volume, Issue, 2002, Pages 575 - 578

[7] KARP Brad, KUNG H. T., "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", MobiCom 2000

[8] HE Tian, STANKOVIC A. John, LU Chenyang, ABDELZAHER F. Tarek, "A Spatiotemporal Communication Protocol for Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed System, Vol. 16, No. 10, October 2005, Pages 995-1006

[9] SUNDARESAN Karthikeyan, ANANTHARAMAN Vaidyanathan, HSIEH Hung-Yun, SIVAKUMAR Raghupathy, "ATP: A Reliable Transport Protocol for Ad-hoc Networks", Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing, 2003, Pages 64 - 75

[10] LU Chenyang, BLUM M. Brian, ABDELZAHER F. Tarek, STANKOVIC A. John, HE Tian, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks", Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'02), IEEE Computer Society, 2002, Pages 55-66

[11] ZHANG Yuecheng, CHENG Liang, "Cross-Layer Optimization for Sensor Networks", New York Metro Area Networking Workshop 2003, New York City, September 12, 2003

[12] AKYILDIZ F. Ian, KASIMOGLU H. Ismail, "A Protocol Suite for Wireless Sensor and Actor Networks", Radio and Wireless Conference, 2004 IEEE, Pages 11 - 14

[13] AKYILDIZ F. Ian, KASIMOGLU H. Ismail, "Wireless Sensor and Actor Networks: Research Challenges", Elsevier: Ad-Hoc Networks, Volume 2, Issue 4, 2004, Pages 351 – 367

[14] LIU W.S. Jane, "Real-Time Systems", Prentice Hall, 2000

[15] WANG Shenquan, NATHUJI Ripal, BETTATI Riccardo, ZHAO Wei, "Providing Statistical Delay Guarantees in Wireless Networks", Proceeding of the 24th International Conference on Distributed Computing Systems, 2004, Pages 48-55

[16] AKYLDIZ F. Ian, SU Weilian, SANKARASUBRAMANIAN Yogesh, CAYIRCI Erdal, "Wireless Sensor Networks: a Survey", Computer Networks Journal (Elsevier), Volume 38, Issue 4, 2002, Pages 393-422

[17] ABDELZAHER F. Tarek, PRABH Shashi, KIRAN Raghu, "On Real-Time Capacity Limits of Multihop Wireless Sensor Networks", Proceedings of the 25th IEEE Real-Time Systems Symposium (RTSS 2004), IEEE, 2004, Pages 359-370

[18] KLEINROCK Leonard, Queueing Systems, New York: Wiley, 1975-1976, vol.1: Theory, vol.2: Computer applications.

[19] MANN-RUBINSON C. Teresa, TERPLAN Kornel, "Network Design Management and Technical Perspectives", CRC Press, 1999

[20] TANENBAUM S. Andrew, "Computer networks", Prentice Hall, 2003

[21] HAYES F. Jeremiah, BABU Ganesh, THIMMA V.J., "Modeling and Analysis of Telecommunications Networks", A Wiley-Interscience Publication, 2004

Annex 1

Priority Function with Four Values

In the figure 4.19 is presented the four evaluation zones for the priority function.

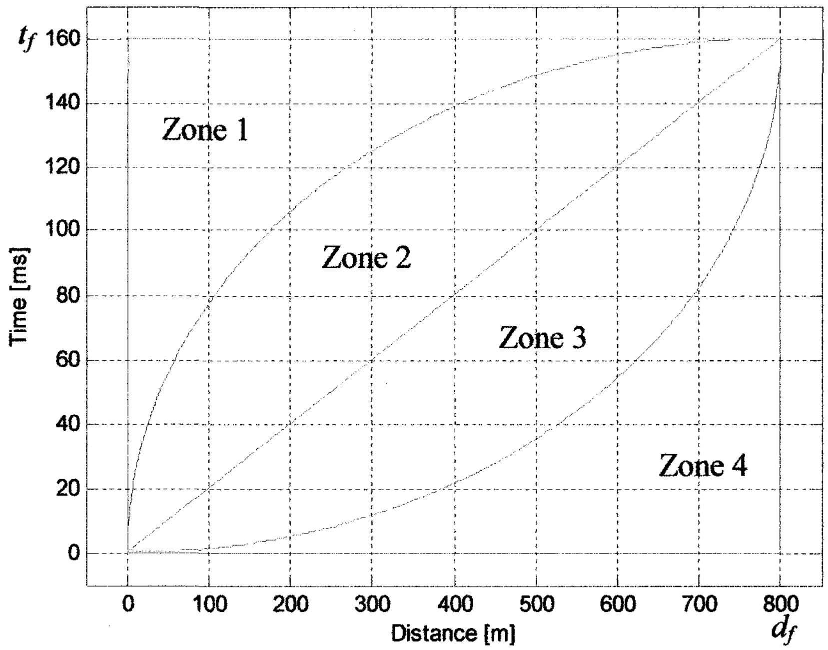


Figure 4.19 Evaluation zones for priority function ζ, with four values

The first zone is delimited by the curves: $d = 0$; $t = t_f$ and $t = \frac{t_f}{d_f}\sqrt{d_f^2 - (d - d_{f)})^2}$ where d_f is the entire distance to travel and t_f the deadline. If a data unit is evaluated to be in this zone, priority the function takes the value "1" and the data unit will have the highest priority.

The second zone is delimited by the curves $t = \frac{t_f}{d_f}\sqrt{d_f^2 - (d - d_{f)})^2}$ and $t = \frac{t_f}{d_f} d$ .The priority function takes the value "2".

The third zone is delimited by the curves: $t = \frac{t_f}{d_f} d$ and $t = t_f - \frac{t_f}{d_f}\sqrt{d_f^2 - d^2}$. The priority function takes the value "3".

The fourth zone is delimited by the curves: $t = t_f - \frac{t_f}{d_f}\sqrt{d_f^2 - d^2}$, $t = 0$ and $d = d_f$. The priority function takes the value "4" and the data units that receive this value have the lowest priority.

In the figure 4.20 shows the priority function ζ that takes four values and utilizes the zones described above.

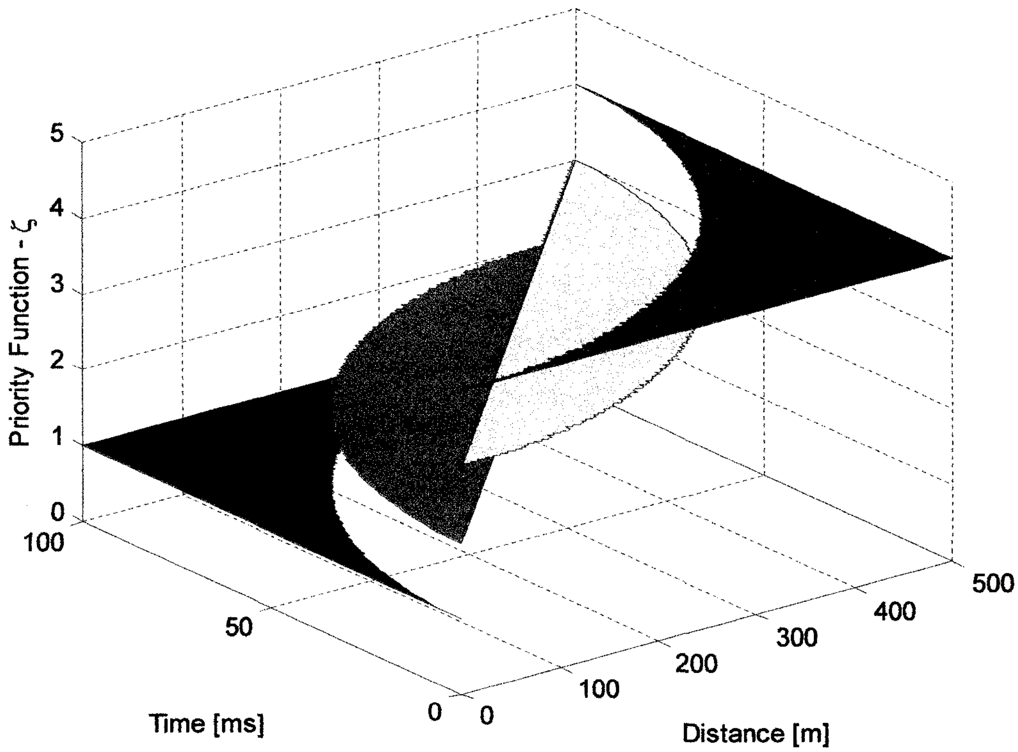


Figure 4.20 The priority function ζ, with four values

Annex 2

Sensor Network with Middle Transmission Rate

We have chosen the transmission rate of W = 512 kbit/s (64 kbyte/s) for this scenario. In the figures 4.21 and 4.22 there are compared the two protocols timelyputs for a different range of deadlines. In the figure 4.21 the deadline range values are between 0 and 400 ms and figure 4.20 presents the .variation of the timelyput in the interval 0-100 ms of the deadline. Up to the value of 40 ms the difference between the two protocols is not important. The difference became more and more important between 40 – 150 ms. The timelyput values remain constant after the value of the 150 ms. Starting with this deadline value 150 ms the protocols difference timelyput is about 200,000 units. BVP behaves better than UDP with about 18%.

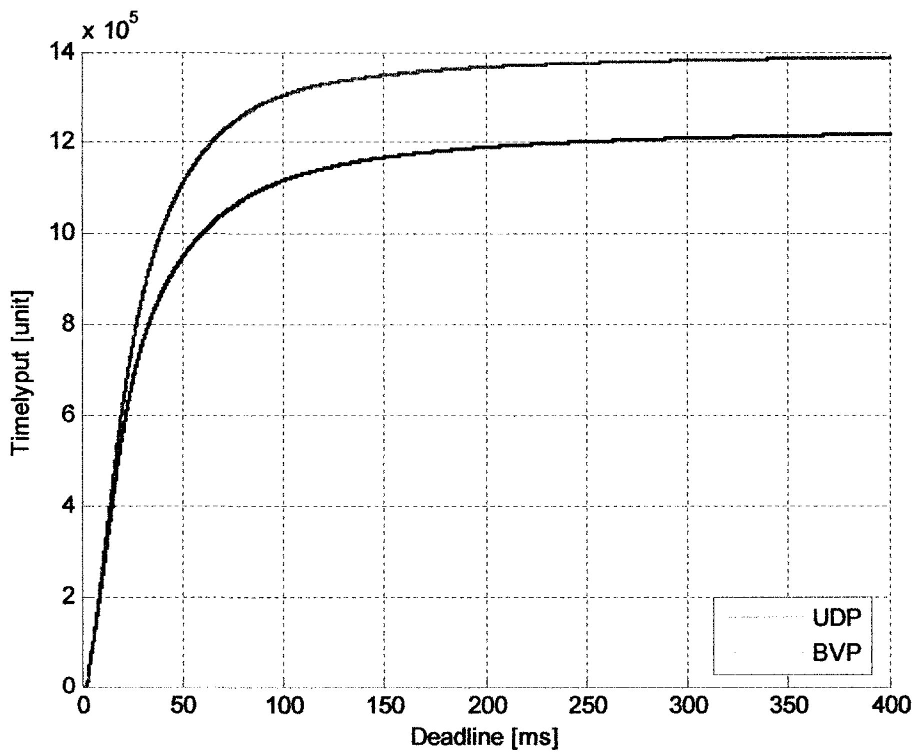


Figure 4.21 Protocol timelyputs versus a large deadline interval for middle transmission rate

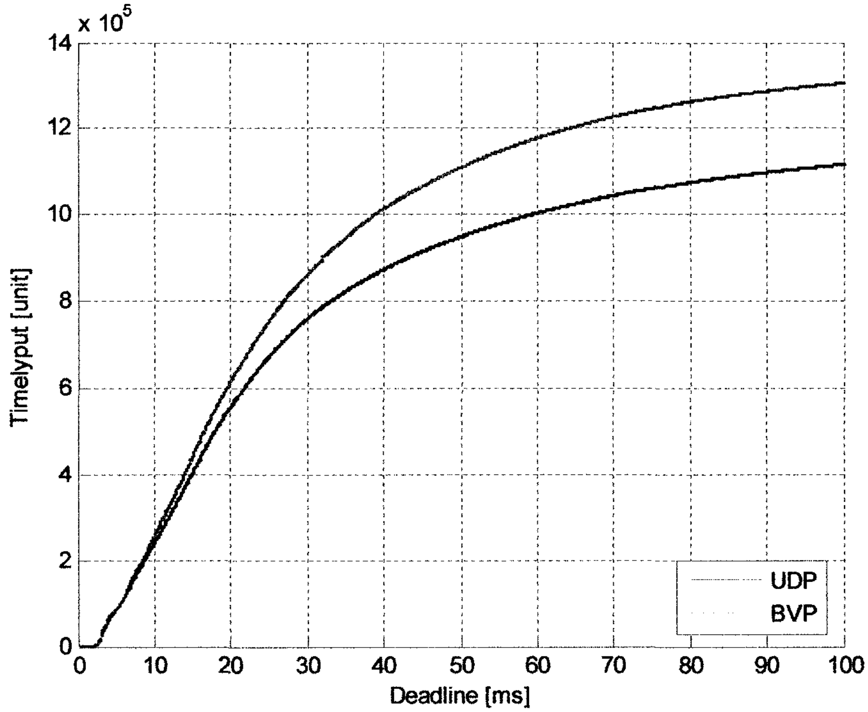


Figure 4.22 Protocol timelyputs versus a short deadline interval for middle transmission rate

In the figures 4.23, 4.24 and 4.25 there are drawn the surface of the variation of the timelyput versus deadline and data unit generation rate for Bonaventura protocol (fig. 4.23), UDP (fig. 4.24) and the difference timelyput between the two protocols (fig 4.25). The deadline interval is 0 – 80 ms and the data units generation rate interval is 0 – 60 data units/s. When the deadline value is large (about 80 ms) and the data unit generation rate is very low, the two protocols have almost the same behavior: the timelyput is about 2,500,000 units which correspond at 50% of the total throughput of the network. As the data generation rate increases, the timelyput of the two protocols decreases but the difference between protocols increases as can be seen on the figure 4.25. When the data unit generation rate arrives at the value of 40 data units/s the network is in a total congestion and the UDP protocol cannot carry any data unit with a travel time less than 80 ms. Under those conditions, BVP protocol has a timelyput of 400,000 units which represent 8% of the total generation number of data units. When the deadline value decreases, the timelyput difference decreases in a linear manner and, for very little values of deadline, the difference is zero.

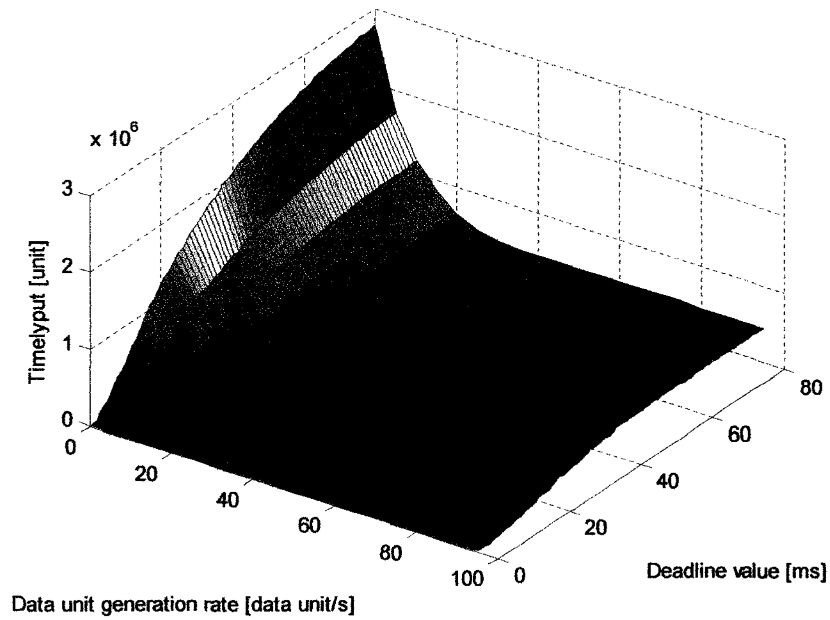


Figure 4.23 BVP timelyput surface for middle transmission rate

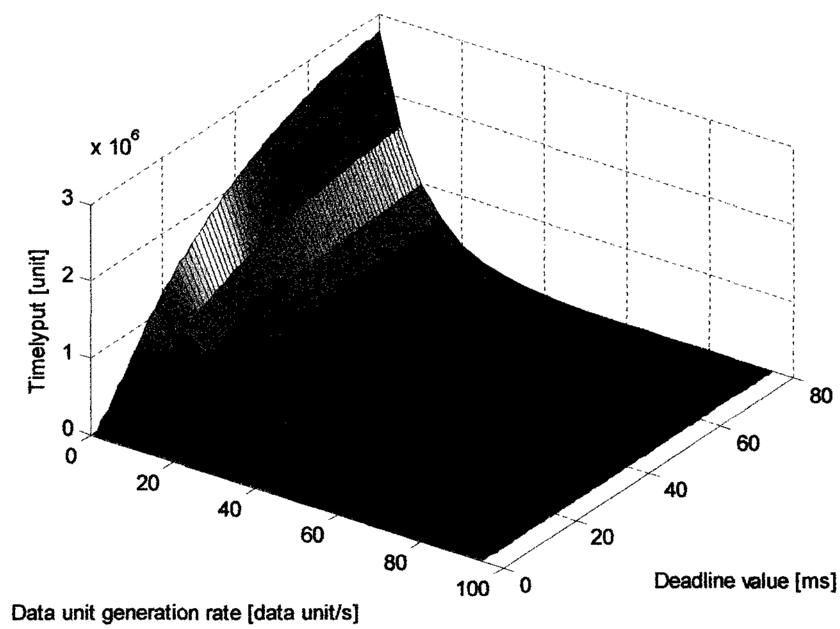


Figure 4.24 UDP timelyput surface for middle transmission rate

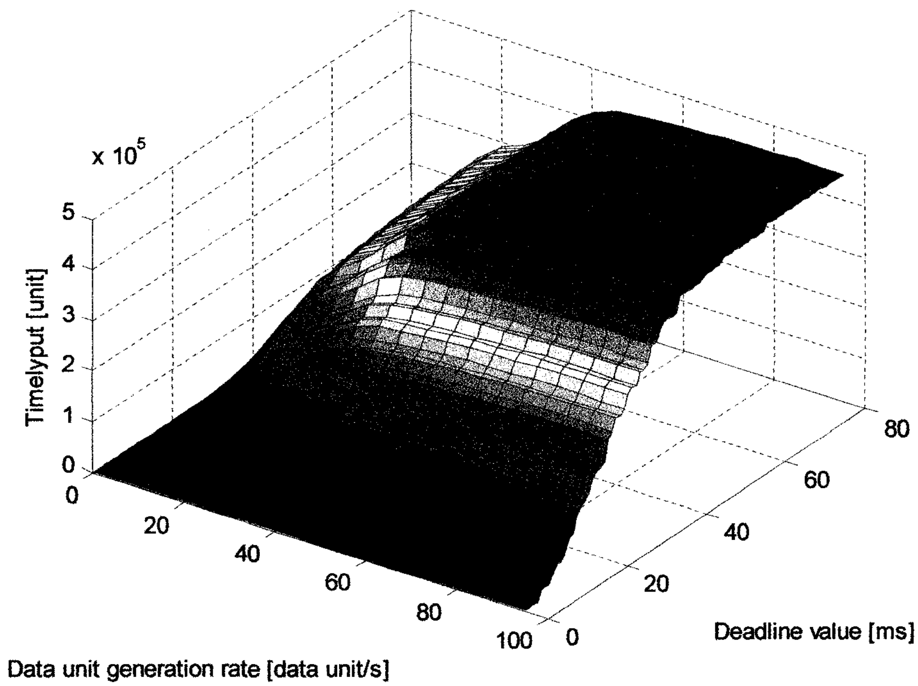


Figure 4.25 Timelyput difference surface for middle transmission rate

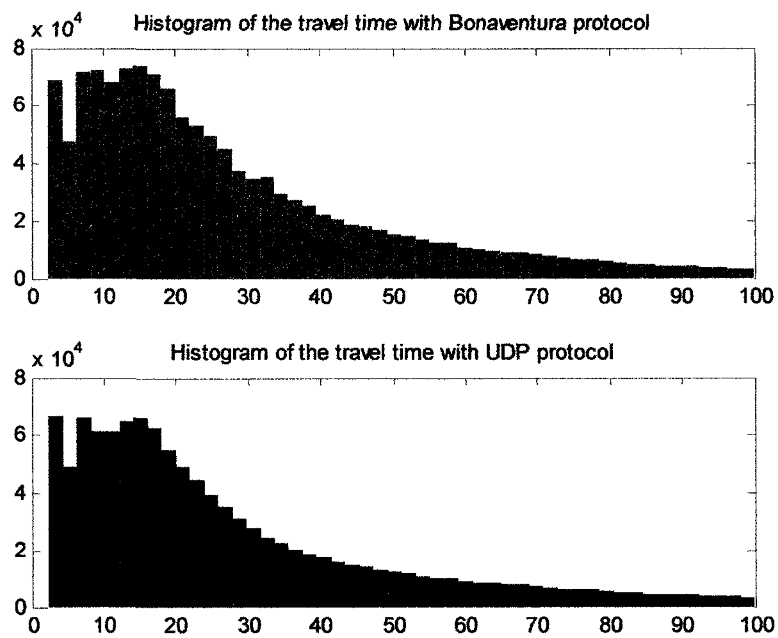


Fig.4.26 Histograms for data units travel times for a middle transmission rate

In the figure 4.26, two histograms are presented, to compare data units travel time of the two protocols. The histograms take the interval 0 – 100 ms and divide it in interval of 2 ms. For each interval there are counted the number of the data units. As it can be seen the number of data units in each interval is greater for BVP than UDP. The most important difference between these protocols is in the interval 6 – 16ms.

As a conclusion, the behaviors of the two compared protocol in a network with middle transmission capacity are analogues as for a networks with low or high transmission capacity. The transmission rate capacity $W = 512$ kbit/s determines middle travel times for data units (as the transmission time for a single data unit is 2 ms) and it also allow middle values of the data unit generation rate. At a data unit generation rate of 40 data units/s the network is in total congestion. The difference between the two protocols is small when the network resources are not exceeded, but this difference increases when the network is used at full capacity.